NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

THE SYSTEMS INTEGRATION OF AUTONOMOUS BEHAVIOR ANALYSIS TO CREATE A “MARITIME SMART ENVIRONMENT” FOR THE ENHANCEMENT OF MARITIME DOMAIN AWARENESS

by

Cledo L. Davis

June 2010

Thesis Co-Advisors: Rachel Goshorn
Deborah Goshorn

Approved for public release, distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2010	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE The Systems Integration of Autonomous Behavior Analysis to Create a "Maritime Smart Environment" for the Enhancement of Maritime Domain Awareness	**5. FUNDING NUMBERS**
6. AUTHOR(S) Cledo L. Davis	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government IRB Protocol number _____N/A________.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Maritime Domain Awareness (MDA) is a very challenging mission area in an ever-increasing net-centric environment, which is inundated with data from many highly advanced, capable sensors and communication suites. With all these technological data collection and dissemination advances, the information available is just too voluminous for humans alone to process and react to manually, sifting the "wheat from the chaff," and be expected to accomplish effective operational decision making regarding maritime threats to national security, as well as to international peace and trade on the high seas.

This thesis addresses MDA Joint Integrating Concept capability gaps, MDA-003C and MDA-004C, for aggregating, analyzing and displaying maritime information in order to understand the maritime environment to identify threats and predicting activity within the maritime domain. Applying the Systems Engineering process, the concept, requirements analysis, architectures, and system design and validation description for a systems integration solution is presented. The proposed implementation entails integrating autonomous behavior analysis capability that utilizes syntactical grammar-based spatial-temporal behavior classifications within existing Net-Centric MDA environments.

In attestation to this implementation, this thesis describes the research conducted on a demonstrable proof-of-concept laboratory system, the Watchman Maritime Smart Environment System, whose representative architecture for specific autonomous behavior analysis implementation is provided.

14. SUBJECT TERMS: Anomaly Detection, Artificial Intelligence, Automation, Behavior Analysis, Distributed Artificial Intelligence, Intelligence-Surveillance-Reconnaissance, Maritime Domain Awareness, Maritime Force Protection, Multi-agent Systems, Network-centric Operations, Network-centric Systems Engineering, Network-centric Warfare, Smart Sensor Networks, Systems Engineering, Systems Integration, System of Systems	**15. NUMBER OF PAGES** 241
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

THE SYSTEMS INTEGRATION OF AUTONOMOUS BEHAVIOR ANALYSIS TO CREATE A "MARITIME SMART ENVIRONMENT" FOR THE ENHANCEMENT OF MARITIME DOMAIN AWARENESS

Cledo L. Davis
Commander, United States Navy
B.S., United States Naval Academy, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2010

Author: Cledo L. Davis

Approved by: Rachel Goshorn
Thesis Co-Advisor

Deborah Goshorn
Thesis Co-Advisor

Clifford A. Whitcomb
Chairman, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Maritime Domain Awareness (MDA) is a very challenging mission area in an ever-increasing net-centric environment, which is inundated with data from many highly advanced, capable sensors and communication suites. With all these technological data collection and dissemination advances, the information available is just too voluminous for humans alone to process and react to manually, sifting the “wheat from the chaff,” and be expected to accomplish effective operational decision making regarding maritime threats to national security, as well as to international peace and trade on the high seas.

This thesis addresses MDA Joint Integrating Concept capability gaps, MDA-003C and MDA-004C, for aggregating, analyzing and displaying maritime information in order to understand the maritime environment to identify threats and predicting activity within the maritime domain. Applying the Systems Engineering process, the concept, requirements analysis, architectures, and system design and validation description for a systems integration solution is presented. The proposed implementation entails integrating autonomous behavior analysis capability that utilizes syntactical grammar-based spatial-temporal behavior classifications within existing Net-Centric MDA environments.

In attestation to this implementation, this thesis describes the research conducted on a demonstrable proof-of-concept laboratory system, the Watchman Maritime Smart Environment System, whose representative architecture for specific autonomous behavior analysis implementation is provided.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ABA – Automated Behavior Analysis

AI – Artificial Intelligence

AIS – Automated Information System, Automated Identification System

ASU – Automated Scene Understanding

ASUW – Anti-surface Warfare

ASW – Anti-submarine Warfare

BAM – Behavior Analysis Module

BCM – Behavior Classification Module

BIT – Built-In Test

C^2 – Command and Control

C4I – Command, Control, Communications, Computers, Intelligence

CCSM – Command And Control Systems Module

CEP – Circular Error Probability

CMA – Comprehensive Maritime Awareness

COCOM – Combatant Commander

COI – Contact of Interest

COMINT – Communications Intelligence

CONOPS – Concept of Operations

COP – Common Operational Picture

COTS – Commercial, Off The Shelf

CPU – Central Processing Unit

CSG – Carrier Strike Group

DHS – Department of Homeland Security

DIPR – Detect, Identify, Predict, React

DML – Daily Muster List

DoD – Department of Defense

DoDAF – Department of Defense Architecture Framework

DRM – Design Reference Mission

ECCM – Electronic Counter-Counter Measures

ECM – Electronic Countermeasures

ECO – Engineering Change Order

ECP – Engineering Change Proposal

EHF – Extremely High Frequency

ELINT – Electronic Intelligence

EO – Electro-Optical

ESM – Electronic Support Measures

EW – Electronic Warfare

FAC – Fast Attack Craft

FIAC – Fast Inshore Attack Craft

FLIR – Forward-Looking Infrared

FOM – Figure of Merit

FSM – Finite State Machine

GCCS-M – Global Command and Control System – Maritime

GENSER – General Service (As to security classification level)

GMTI – Ground Moving Target Indicator

GUI – Graphical User Interface

I&W – Indications and Warning

ID – Identification

IFF – Identify Friend or Foe

IP – Internet Protocol

IPOM – Infer Predicted Outcomes Module

IR – Infrared

ISAR – Inverse Synthetic Aperture Radar

ISR – Intelligence, Surveillance, and Reconnaissance

IUSS – Integrated Undersea Surveillance System

JCA – Joint Capability Area

JCTD – Joint Capability Technology Demonstration

JHOC – Joint Harbor Operation Center

JIC – Joint Intelligence Center

JTAA – Joint Test and Assessment Activity

JWICS – Joint Worldwide Intelligence Communications System

LFA – Low Frequency Array

LLTV – Low-light television

LMRS – Long-term Mine Reconnaissance System

MDA – Maritime Domain Awareness

MDT – Muster Data Tag

MET – Mission Essential Tasks

METL – Mission Essential Task List

MLS – Multi-Level Security

MOE – Measure of Effectiveness

MOP – Measure of Performance

MPA – Maritime Patrol Aircraft

MSC – Maximum Similarity Classification

MSE – Maritime Smart Environment

NMETL – Naval Mission Essential Task List

NPS – Naval Postgraduate School

NTTL – Navy Tactical Task List

OED – OSIS Evolutionary Development

OPORDS – Operational Orders

OPSIT – Operational Situation

OSA – Open System Architecture

OSD – Office of the Secretary of Defense

OSINT – Open Source Intelligence

OSIS – Ocean Surveillance Information System

OTH-T – Over the Horizon Targeting

PANDA – Predictive Analysis for Naval Deployment Activities

PC – Personal Computer

PDM – Position De-confliction Module

PIM – Position Integration Module

POCS – Proof of Concept System

R&D – Research and Development

ROE – Rules of Engagement

SA – Situational Awareness

SAG – Surface Action Group

SAR – Synthetic Aperture Radar

SDD – System Description Document

SDI – System Display Interface

SE – Systems Engineering

SHF – Super High Frequency

SIGINT – Signals Intelligence

SLOC – Sea Lines (or Lanes) of Communication

SME – Subject Matter Expert

SOA – Service-Oriented Architecture

SOSUS – Sound Surveillance System

SQL – Structured Query Language

SSBC – Sequential Syntactical Behavior Classifier

Sub HDR – Submarine High Data Rate

SURTASS – Surveillance Towed Array Sonar System

T-AGOS – Tactical Auxiliary General Ocean Surveillance

TACTAS – Tactical Towed-Array Sonar

TC – Tolerance Comparator

TPM – Technical Performance Measure

TPPU – Task, Post, Process, and Use

TS/SCI – Top Secret / Sensitive Compartmented Information

UAS – Unmanned Arial System

UAV – Unmanned Arial Vehicle

UJTL – Universal Joint Task List

UNTL – Universal Naval Task List

USB – Universal Serial Bus

UUV – Unmanned Underwater Vehicle

VDI – Video Display Interface

WAES – Watchman Applications Engineering Subsystem

WMA – Warfighting Mission Areas

WMSE – Watchman Maritime Smart Environment

WNES – Watchman Network Engineering Subsystem

WSES – Watchman Software Engineering Subsystem

WSSN – Wireless Smart Sensor Network

EXECUTIVE SUMMARY

National Security Presidential Directive NSPD-41 / Homeland Security Presidential Directive HSPD-13 lays out the Maritime Security Policy for the United States. In that document, the U. S. President stated, “It is critical that the United States develop an enhanced capability to identify threats to the Maritime Domain as early and as distant from our shores as possible…” It is just this type of capability that the President claimed the United States must have that this thesis addresses. This thesis explores the Systems Engineering challenge of meeting the stated mission needs of enhancing the capability to identify threats in the maritime domain through the employment of the Systems Engineering process as applied to a specific research project within the Network-Centric Systems Engineering Track of the Systems Engineering Department of the Naval Postgraduate School.

This thesis presents how Maritime Domain Awareness, or MDA, can be enhanced through the application of SE to create what will be described as a “Maritime Smart Environment” (MSE). This MSE capability is created through the integration of an “Automated Behavior Analysis” (ABA) capability, or system, into the current typical maritime network of platforms, sensors, and data fusion applications (also known as the *MDA network*) found in a U.S. Navy Carrier Strike Group (CSG) or Surface Action Group (SAG). A Design Reference Mission (DRM), created for this thesis, defines the capabilities, concept of operations (CONOPS), and mission for this type of MDA network, and provides the scope, boundary conditions, and context for the clear understanding of what MDA capability enhancement the notional MSE, integrating the ABA, is intended to achieve.

This thesis describes the research associated with the Watchman Maritime Smart Environment Proof Of Concept System (WMSE POCS) project, along with the SE processes used to implement and demonstrate it, in order to establish the feasibility of integrating an ABA into an existing Net-Centric environment within the Maritime Domain. The integration of this capability, with its demonstrated potential to recognize particular behaviors and alert operators to them, reveals the opportunity to further develop this technology and systems integration concept to fill sorely needed MDA Joint Integrating Concept (JIC) capability gaps of aggregating, displaying, and analyzing maritime information in order to understand the maritime environment and identify threats (MDA-003C) and predicting activity within the maritime domain (MDA-004C). Meeting the numerous MDA mission requirements that these gaps represent will most assuredly enhance Maritime Domain Awareness by improving situational awareness, better utilizing the capability of networked sensors and the multitude of data they provide, as well as making operators more efficient in their duties to respond to real threats, rather than ineffectively attempting to manually monitor and react to threats arising from the whole of the maritime domain.

After a brief introduction explaining the operational concept and mission need germane to the thesis, the development of the case begins with an explanation of the current MDA capabilities and practice with a survey of the various representative platforms and sensor systems used to accomplish MDA in the different media within the maritime environment, and then takes a brief look at some of the current R&D initiatives for MDA enhancement that incorporate or attempt to utilize some form of automation.

The thesis then discusses the SE approach in the conduct of the actual research, as well as how this approach, consisting of a concept and process, could be used as a model for physical implementation of real-world mission-capable ABA. The next chapter expounds upon a key part of the SE process by describing the development of the WMSE POCS system architecture, showing the functional, process, and operational decompositions for the initial system, as well as how subsequent iterations of the requirements and functional allocations for system upgrades produced refined software architecture, as well as an integration architecture. The following chapter thoroughly describes the WMSE POCS, from initial concept to latter upgrades, additional capabilities, and improvements, which demonstrated its proven functionality in analyzing and recognizing behaviors in a network-centric engineered laboratory environment. The latter portion of this chapter addresses operational implementation of the POCS, by describing the two types of operational implementation System Engineers could expect to encounter when attempting to implement an ABA capability. These alternatives were analyzed, with the resultant recommendation of the singular integration of an ABA capability into the existing NCW environment, as the most cost-effective and mission-capable implementation solution.

The task of monitoring the maritime domain and remaining vigilant to the threats to our national security that emanate from it will remain a vital mission area for any foreseeable future, exemplified by the MDA JIC capability gaps addressed in this thesis. The maritime domain is an ever-increasing net-centric environment, with a continually growing number of sensors, operating nodes, communications links, providers, and consumers of information. With these growing numbers of both fixed and mobile capabilities, there will continue a deficit in humans, bandwidth, power, and intelligent centers to deal with this cascade of data. There can be no doubt that MDA is a very challenging field, with the majority of the earth's surface falling under its purview. Highlighted in this research are the many advances that have been made in the field, in the way of more highly capable, longer range, and extended duration sensors; more robust and reliable communications; higher bandwidth and faster networks with improved data sharing; and lastly much higher fidelity and accuracy in the fusion of the sensor data which is passed among these networks. With all these advances, the data available is just too voluminous for humans alone to process and react to manually, sifting the "wheat from the chaff," and be expected to accomplish effective operational decision making regarding maritime threats to national security and international peace and trade.

The ability of integrating an ABA capability through scaled development, as presented in this thesis, would certainly enhance the application of MDA through fulfillment of stated JIC capability gaps. This integration methodology, put into operational practice, would allow a more seamless implementation of maritime prediction capability, less disruption of current sensor data collection capability, processes, and operations, and ultimately a satisfaction of the critical mission needs that exist in maritime situational awareness, threat recognition and response.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to extend my most heartfelt gratitude to my two Co-Advisors, Rachel and Deborah Goshorn for their tireless efforts in guiding this research project to fruition. Without their keen insight, endless encouragement and abounding spirits, this thesis would not have been possible.

Thanks go to Professor Gary Langford for his unique and insightful guidance, which proved invaluable, and for always being willing to make time for my many questions and me.

Thanks also to a super research assistant, Wenonah Hlavin. Your assistance in getting me the help I needed in CORE (ugh!), and providing spot-on references when I needed them was phenomenal.

I want to thank my awesome kids, Clay II, Rebecca, Benjamin, Joshua, and Sarah. You guys are my biggest fans and having you guys "in the stands" cheering me on, and always praying "Dear God, Please help Daddy to get an A-plus-plus on his paper" were the inspiration that kept me going when I wanted to give up. I'm so proud of you and love you all more than you will ever know.

And to Loralee, the best Navy wife anyone could ever want or ask for; you have been right there by my side, keeping the home fires burning, while supporting, encouraging, and loving me every step of this journey. We knew it wouldn't be easy, but now this chapter of our life has closed, and a new one is just beginning. I am more than blessed to have you as my bride. I love you, Darlin'.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The introduction chapter discusses the overview of the thesis, a description of the Design Reference Mission (DRM), the thesis research approach, and the outline of the thesis. The overview section includes the rationale for and relevance of the thesis. The DRM, its purpose and components, is explained in detail, as it is referenced throughout the thesis. The thesis research approach introduces the systems engineering process, as well as its application in the proof of concept system, which is the focus of the research. Lastly, the thesis outline briefly describes the content and purpose of each chapter in the thesis to guide the reader through the document.

A. OVERVIEW

National Security Presidential Directive NSPD-41 / Homeland Security Presidential Directive HSPD-13 lays out the Maritime Security Policy for the United States. In that document, the president stated, "It is critical that the United States develop an enhanced capability to identify threats to the Maritime Domain as early and as distant from our shores as possible…" This type of needed capability declared by the president is what this thesis addresses. The directive goes on to note, "Due to its complex nature and immense size, the Maritime Domain is particularly susceptible to exploitation and disruption by individuals, organizations, and States." To counter this disruption and nefarious use of the Maritime Domain by "terrorists, criminals, and hostile States," the president directed that, "The United States must deploy the full range of its operational assets and capabilities," in accomplishing the task.[1] Understanding how capabilities needed to meet certain tasks become requirements, and then further understanding how those requirements must be refined and turned into a producible, effective, and affordable design solution is the purpose of Systems Engineering (SE).[2] This thesis explores the

[1] National Security Presidential Directive NSPD-41 / Homeland Security Presidential Directive HSPD-13, December 2004.

[2] Benjamin S. Blanchard and Wolter J. Fabrycky, *Systems Engineering and Analysis*, 17–18, Englewood Cliffs, NJ: Prentice-Hall, 2006.

Systems Engineering challenge of meeting the stated mission needs of enhancing the capability to identify threats in the maritime domain through the employment of the Systems Engineering process as applied to a specific research project within the Network-Centric Systems Engineering Track of the Systems Engineering Department of the Naval Postgraduate School.

This thesis shows that Maritime Domain Awareness, or MDA, can be enhanced through the application of SE to create what will be described as a “Maritime Smart Environment” (MSE). This MSE is created through the integration of an “Automated Behavior Analysis” (ABA) capability, or system, into the current typical maritime network of platforms, sensors, and data fusion applications (also known as the *MDA network*) found in a U.S. Navy Carrier Strike Group (CSG) or Surface Action Group (SAG). A Design Reference Mission (DRM), created for this thesis, will define the capabilities, concept of operations (CONOPS), and mission for this type of MDA network, and will provide the scope, boundary conditions, and context for the clear understanding of what MDA capability enhancement the notional MSE, integrating the ABA, is intended to achieve.

B. DESIGN REFERENCE MISSION (DRM) DESCRIPTION

Defining a DRM is one of the primary pieces of the requirements analysis and allocation phase of the SE process. The entire SE process is discussed in more detail in Chapter III. In order to accomplish effectual systems engineering, accurate problem definition is required to ensure appropriate problem solving. A predetermined problem solution or a solution that attempts to solve the wrong problem may very well result in a biased and/or flawed design. Defining the problem via the generation of mission needs from determined capability gaps will ultimately lead to a more accurate problem solution that will be much more likely to support the entire range of stakeholders’ needs.

The complexity and broad scope of Maritime Domain Awareness makes establishing the specific threat type and environment in addressing a particular capability gap vital. The DRM concept is used in order to establish a war fighting CONOPS for a

MDA enhancement baseline context. A DRM defines the operational activities necessary to validate required capability attainment.

> The DRM establishes the baseline for subsequent systems engineering activities—particularly generation of requirements, refining problem definition, development of concepts, and analysis of alternatives, and testing and evaluation. A well-developed DRM will facilitate generation of requirements and subsequent system design.[3]
>
> For the government led development process, the DRM feeds the development and certification of a system functional baseline and provides support through the entire life of the program. Thus the DRM must support the program throughout the systems engineering process.[4]

Designing a reference mission begins with understanding the environment surrounding the mission needs analysis. The DRM will have a scenario that includes a goal, a threat, a deployment of systems, a physical environment in which the mission takes place or is executed, and whatever changes the environment may go through as the scenario progresses.

1. Problem Definition

To protect our vital national interests, high-value assets at sea must be protected, and are therefore often nested within a network of internal and external, organic and inorganic, manned and unmanned, sea-based and land-based sensors that attempt to provide a Common Operational Picture (COP) of the sea surface around the high-value unit(s). These types of sensor networks increasingly have the ability to fuse much, if not all, of the sensors' data flooding in to detect, track, and, to a certain extent, classify the surface vessels in the vicinity of these unit(s). Currently, humans in the loop of these sensor networks monitor the contact tracks of these vessels; however, their dense number and irregular, indiscernible, and unpredictable movements prevent an early and/or accurate detection of threat-like activity by the many contacts that must be monitored in the COP. Automation of these vital, yet mundane, monitoring and threat-detection

[3] F. R. Skolnick and P. G. Wilkins (2000), "Laying the foundation for successful systems engineering," *Johns Hopkins Apl Technical Digest 21*, no. 2 (2000).

[4] Ibid.

activities could potentially enhance greatly the awareness of the maritime domain for the protection of friendly forces. For example, among the many fishing and commercial vessels going about their normal routines and other various surface traffic, networked sensors may pick up a group of vessels moving in some type of formation, apparently approaching the high-value assets. Watch-standers who fail to detect this activity or assume that this behavior is no threat due to its indiscernible nature, could delay critical seconds in responding, which could lead to a devastating loss of ship and crew. On the other hand, assuming this to be an imminent threat and taking lethal action too quickly could cause an adverse international incident if incorrect. It is just this type of scenario that reveals the gap in current capability, which is in providing decision makers with the crucial pieces of timely information to discern truly hostile from innocent behavior out of the myriad pieces of data available, in order to be able to take appropriate action.

2. Projected Operational Environment (POE)

The POE provides information for establishing a context within which operational tasks will produce their functional outcomes. This context is defined by the desired capabilities from which the functional outcomes and requisite tasks are derived. These desired capabilities are provided via an analysis of the various national security directives and the Joint Operations Guidance (JOG) that flows from them. The Joint Staff is tasked with integrating these capabilities across the services and all lines of operation, under the JOG, to determine where, operationally, gaps in the desired capabilities exist. From these gaps, services then develop requirements, which are then passed on to the systems commands to procure solutions to meet those requirements.[5] In addressing these gaps, the Systems Engineers must analyze the requirements as the initial step of the Systems Engineering Process (which is discussed in detail in later chapters) to obtain the optimal solution to meet the operational need.

[5] CJCSI 3010.02B, Joint Operations Concepts Development Process (JOpsC-DP), January 27, 2006.

From this process, the eventual solution (notionally some type of Net-Centric Maritime Smart Environment (MSE)) is intended to meet several DoD MDA Joint Integrating Concept (JIC) capability gaps, listed below:[6]

- MDA-003C — The capabilities to aggregate, display, and analyze maritime information in order to understand the maritime environment and identify threats. The purpose of this capability is to make maritime information available to the MDA network in a useful form. This involves fusing and aggregating data so that users can quickly retrieve all of the information related to an entity of interest. The MDA network must allow analysts to easily manipulate information and be flexible enough that they can use their own creativity to detect latent patterns of threat behavior.
- MDA003C-003T — Provide alerts for suspicious vessel behavior in particular for small non-emitting vessels.
- MDA-003C-006T — Archive and make available historical information. One important purpose is to support post-event analysis for attack attribution.
- MDA-003C-007T — Push time-critical alerts to decision makers
- MDA-003C-008T — Analyze all source information to determine which cargos are high risk or of interest
- MDA-003C-009T — And analyze all source information to determine which persons are high risk or of interest
- MDA-003C-010T — Analyze all source information to characterize threat networks
- MDA-004C — The capability to predict activity within the maritime domain. Many non-traditional threats, like terrorism or proliferation, are

[6] J7, Joint Force Development and Integration Division (JFDID), Maritime Domain Awareness JIC, Version 0.3, January 8, 2009.

effective when they operate within the noise level of normal, everyday activities. Detecting these plots often requires merging information from all aspects of the maritime domain to detect the subtle signs of threat behavior. The ability for machines to automatically analyze mountains of data and detect anomalous behavior will be critical for making sense of the vast maritime realm. Automated threat analysis systems free analysts to work on more challenging maritime problems.

- MDA-004C-001T — Create a baseline of normal maritime behavior for an area or conditions of interest
- MDA-004C-002T — Identify adversary patterns of behavior
- MDA-004C-003T — Differentiate maritime threats from valid maritime commerce
- MDA-004C-004T — Provide alerts for suspicious behavior
- MDA-004C-005T — Identify adversary intent, courses of action, strengths, and weaknesses

3. Potential Friendly Targets

The DRM must define what types of targets the enemy is likely to exploit, in order to design accurate scenarios that properly represent the scoped mission need.

Under this thesis, the MSE would be primarily deployed for force protection and so its employment would not include any offensive scenarios. Some possible defensive scenarios would include:

- Protection of fixed at-sea platforms
- Protection of strike groups or sea bases
- Surveillance of ports or harbors
- Providing freedom of navigation through possible choke points
- ISR missions to locate enemy forces
- Tracking and interception of suspicious vessels

For the scope of this thesis, the DRM will focus only on the protection of strike groups or sea bases scenarios, from which the following Operational Situation (OPSIT) will be derived.

4. Operational Situation (OPSIT)

OPSITs typically detail various cases within a DRM where certain variables can change, creating unique OPSITs for a particular mission scenario. OPSIT outcomes derived from simulations can eventually be compared to actual system test data, verifying that the system's operational activities are sufficient to perform the mission effectively. Input from Subject Matter Experts (SMEs) is imperative for quality OPSIT development. OPSITs should be validated by SMEs, ensuring a balance between common and extreme cases and variables.[7]

The systems engineer must go through a comprehensive planning process to understand "how" the mission will be accomplished. This process is an indispensible part of the architecting activity in the SE process, which is discussed in Chapters III and IV. This mission analysis produces a plan with certain tasks to be assigned to the operational nodes though operational activities in order to complete a mission. A mission typically will consist of multiple operational activities, involving multiple system elements simultaneously conducting a variety of assigned tasks. These tasks must integrated and synchronized in order to accomplish the operational activities necessary to achieve the mission. System architectures detail and decompose the relationships between the operational nodes, operational activities, tasks, system elements, and required system functions. OPSITs are important to this activity by depicting the tasks required to perform the mission commander's CONOPS. The commander must determine the tasks that are essential to mission success and identifies these as Mission Essential Tasks (MET). The MET are typically derived from the Universal Joint Task List (UJTL), Universal Naval Task List (UNTL), Navy Tactical Task List (NTTL) and the Naval Mission Essential Task List (NMETL), as appropriate. During OPSIT

[7] Skolnick and Wilkins, "Laying the foundation."

generation, a set of operational tasks are defined for every operational activity achieved by a mission in order to develop a war fighter CONOPS. Assumptions are developed regarding the environment, logistics, deployment, time, and other factors required to achieve the mission. Assumptions should be realistic variables designed to afford defined parameters for the scenario. The systems engineer must determine what parameters are key to studying system performance and which can be assumed at certain levels.[8]

The specific mission for the OPSIT in this thesis is to counter the threats posed by a "swarm" attack. For several years, the U.S. Navy and several international navies have identified the attack by large groups (swarms) of small attack boats as a major and growing littoral threat. These attack boats, generally identified as "fast attack craft (FAC)" and "fast inshore attack craft (FIAC)" by the U.S. Navy, vary significantly in size, speed, crew, and weaponry; and are widely available to small and developing nations. There have been several U.S. Navy studies and experiments that have concluded that currently deployed naval weapons, sensors, and Command and Control (C^2) systems would have difficulty in effectively and reliably countering large FIAC swarm attacks.[9]

The emergence of large numbers of FIACs and swarm tactics in Iran and other "developing world" countries has highlighted the potential near-term significance of this threat. Thus, there is a requirement to extend the Navy's current surface warfare capability with new defense concepts that will be effective against this established and growing asymmetric threat.

In addition, the other rapidly emerging threat is piracy on the high seas in the shipping lanes near the horn of Africa and elsewhere. The tactical concerns in regards to the FIAC threat discussed above are much the same with the pirate threat, only the potential targets and motives of the pirate threat are quite different from what would be expected in a FIAC attack. The common mode of employment of pirate attack vessels is reported to be larger "mother ships" providing support, command, and control to

[8] Skolnick and Wilkins, "Laying the foundation."

[9] The Office of Naval Research Broad Agency Announcement 09-023, "Multi-Target Track and Terminate (MT3)," 2009.

numerous other small, fast "take-down" vessels that search out and board commercial ships for hijacking and eventual ransom demands.[10]

Regardless of the differences in these types of threats, the ability to quickly and accurately discern threat behavior, of the particular vessels that seek to do harm versus the multitude of other surface traffic around a high-value asset, is required in both cases.

5. Threat Profile

In order to understand the tasks required for the mission defined in the DRM, the type of threat must be understood. This understanding comes from describing the particular profile of the threat that the mission must counter. This profile is made up of the following components: Assumed General Threat Conditions (Peculiar aspects or capabilities of the threat), Threat Approach (where the threat will originate from and how it is assumed to approach friendly forces), Threat Characterization (specific characteristics that the threat may have with a relative probability of exhibiting those characteristics), Threat Tactics, and Attack Timing and Coordination.

The most likely threat for this DRM is an enemy force (conventional, pirate, or terrorist) that can attack by small, fast surface vessels to hijack, damage, or destroy the friendly target. Here following is a delineation of the threat profile elements for this most likely threat scenario.

Assumed Threat General Conditions:

- A reasonably sophisticated terrorist organization that is non-state sponsored
- A suicide force capable of a covert, coordinated surface attack
- A small group of pirates armed with small arms and RPGs
- Attack platforms that are commercially available
- Surface craft similar in size and appearance to indigenous commercial or pleasure boats (but modified for higher performance)

[10] W. Thomas Smith Jr., "Pirates in the 21st Century," July 3, 2006, available at http://www.military.com/forums/0,15240,103960,00.html, accessed May 20, 2010.

- Only conventional explosives and weapons are used
- Minimal early indicators of a pending attack
- Attack occurs at any time of day and the weather conditions are clear with low wind speeds

Assumed Threat Approach:

- Threat surface vehicles transit to the general area of the platform from the direction of a nearby commercial fishing fleet in a boat of similar appearance

Threat Characterization:

Table 1 lists the specific characteristic of speed that the threat may possess with a relative probability of exhibiting that characteristic. Other characteristics, such as varying size, shape, crew size, etc. were considered; however, they were deemed to be non-essential to characterizing the threat profile, due to the negligible effect those characteristics would have on the threat behavior in this scenario, as compared with speed.

Threat	Speed	Probability
Surface Vessel	20 knots	Low
	30 knots	Medium
	40 knots	High

Table 1. DRM Threat Characteristics

Threat Tactics:

Swarm tactics are to conduct a coordinated, simultaneous raid upon their target with a certain number of attack vessels. The number of vessels (*n*) in the raid may vary upon the type of threat (pirate, terrorist, state actor), distance from shore, capabilities, etc.

- Assumed raid size possibilities (number vessels = n)
    - $6 > n < 10$
    - $3 > n < 5$
    - $0 > n < 2$

Attack Timing and Coordination:

Although typical swarm tactics are to conduct simultaneous raids when approaching the target, the actual conduct of the raid as attack vessels engage the target can vary. Possible options include:

- One at a time
- All at once in a concentrated location
- Surround surveillance area and then simultaneous attack

6. Mission Success Requirements

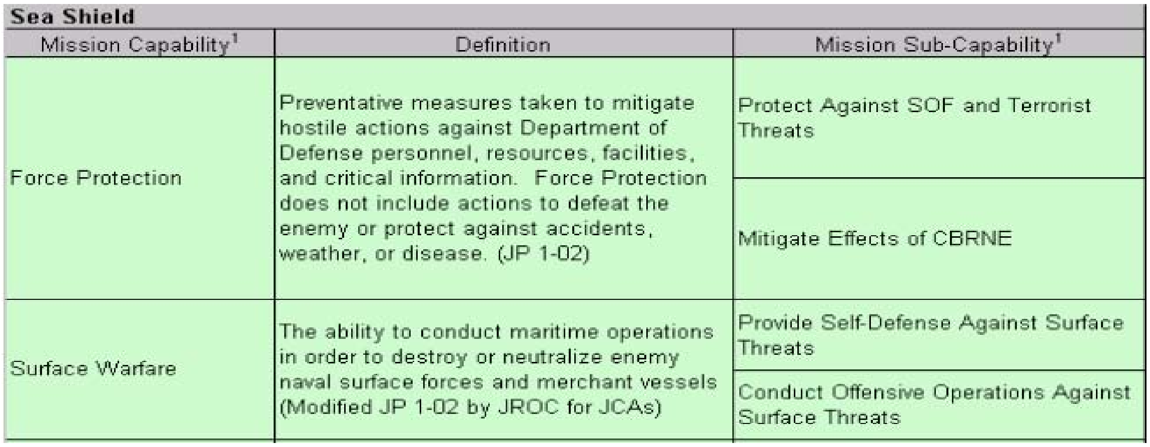
As discussed, the OPSIT identifies the individual activities that need to be accomplished in order to define the success of the mission. The activities identified for the success of this DRM will be evaluated in these categories**:**

- Provide information transfer
- Conduct information processing
- Identify and classify targets
- Provide cueing and targeting info
- Provide accurate behavior classification of surface threats
- Facilitate engagement of time sensitive and non-time sensitive targets

The mission is divided into these categories based on the specific functions that each individual operational activity is required to perform. Each category must be completed in order to identify the mission as being successful.

7. Mission Definition

In order to complete the mission success levels, all operational activities are utilized. Each mission included within a DRM scenario can be decomposed into the individual operational activities necessary to complete the tasks that the DRM scenario requires. The Joint and Naval Capability Terminology List is a compilation of Joint and Navy capabilities areas. The Joint Capability Areas (JCAs) are broken into Warfighting Mission Areas (WMA), which include Joint Training, Command & Control, Force Application, Force Protection, Focused Logistics, Battlespace Awareness and Force Management.[11] The naval capabilities are taken from the U.S. Navy's transformational vision document, Naval Power 21, which is a combination of Sea Power 21 and Expeditionary Maneuver Warfare Capabilities. Naval Power 21 has five pillars, which are Sea Shield, Sea Strike, Sea Basing, Expeditionary Maneuver Warfare, and FORCEnet.[12]



Sea Shield		
Mission Capability[1]	Definition	Mission Sub-Capability[1]
Force Protection	Preventative measures taken to mitigate hostile actions against Department of Defense personnel, resources, facilities, and critical information. Force Protection does not include actions to defeat the enemy or protect against accidents, weather, or disease. (JP 1-02)	Protect Against SOF and Terrorist Threats
		Mitigate Effects of CBRNE
Surface Warfare	The ability to conduct maritime operations in order to destroy or neutralize enemy naval surface forces and merchant vessels (Modified JP 1-02 by JROC for JCAs)	Provide Self-Defense Against Surface Threats
		Conduct Offensive Operations Against Surface Threats

Table 2. Sea Shield Pillar from Naval Power 21 (and Naval Capability Terminology List, 2007) (After NEARG)[13]

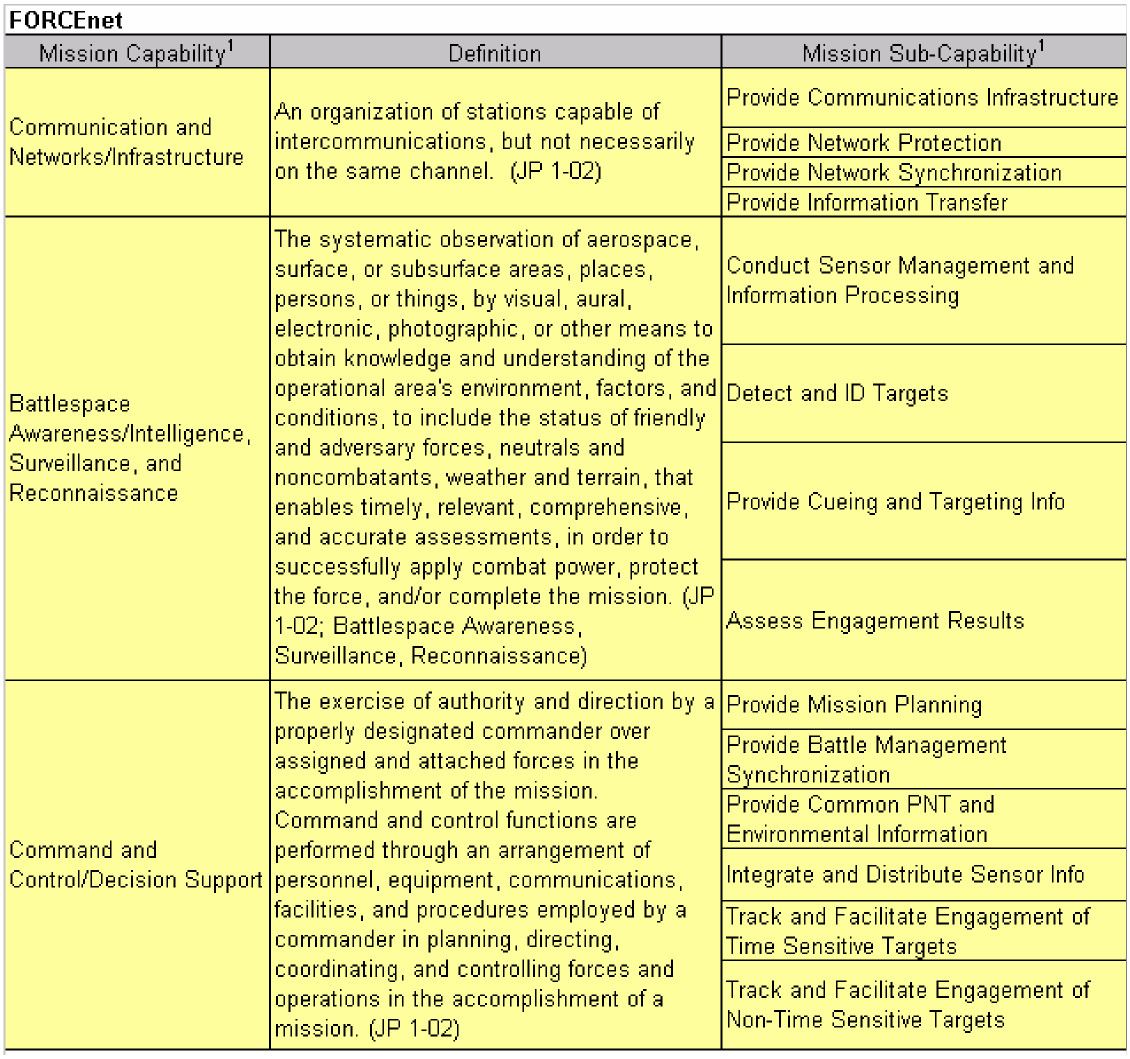
The capabilities focused upon in this case are Sea Shield and FORCEnet, as the capabilities encapsulated in those pillars most closely align with the capabilities required

[11] CJCSI 3010.02B.

[12] "Sea Power 21 – Projecting Decisive Joint Capabilities," Admiral Vern Clark, U.S. Navy, *Proceedings*, October 2002.

[13] Naval Architecture Elements Reference Guide, https://stalwart.spawar.navy.mil/naerg/.

for this DRM, as well as for conducting MDA, in general.[14] The missions within that capability that is focused upon are Force Protection and Surface Warfare, as shown in Table 2, and all mission areas of FORCEnet as shown in Table 3. These focused mission areas were chosen because they specifically address the OPSIT generated under this specifically designed reference mission.



FORCEnet

Mission Capability[1]	Definition	Mission Sub-Capability[1]
Communication and Networks/Infrastructure	An organization of stations capable of intercommunications, but not necessarily on the same channel. (JP 1-02)	Provide Communications Infrastructure
		Provide Network Protection
		Provide Network Synchronization
		Provide Information Transfer
Battlespace Awareness/Intelligence, Surveillance, and Reconnaissance	The systematic observation of aerospace, surface, or subsurface areas, places, persons, or things, by visual, aural, electronic, photographic, or other means to obtain knowledge and understanding of the operational area's environment, factors, and conditions, to include the status of friendly and adversary forces, neutrals and noncombatants, weather and terrain, that enables timely, relevant, comprehensive, and accurate assessments, in order to successfully apply combat power, protect the force, and/or complete the mission. (JP 1-02; Battlespace Awareness, Surveillance, Reconnaissance)	Conduct Sensor Management and Information Processing
		Detect and ID Targets
		Provide Cueing and Targeting Info
		Assess Engagement Results
Command and Control/Decision Support	The exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of the mission. Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of a mission. (JP 1-02)	Provide Mission Planning
		Provide Battle Management Synchronization
		Provide Common PNT and Environmental Information
		Integrate and Distribute Sensor Info
		Track and Facilitate Engagement of Time Sensitive Targets
		Track and Facilitate Engagement of Non-Time Sensitive Targets

Table 3. FORCEnet Pillar from Naval Power 21 (and Naval Capability Terminology List, 2007) (After NAERG)

[14] National Concept of Operations for Maritime Domain Awareness, December 2007.

The Consolidated (formerly called "Common") Operational Activities List (COAL) is part of the Department of Defense Architecture Framework (DoDAF)[15] and is used to create a fully integrated joint operational activities list to support the Combatant Commanders (COCOM's) and architecture development. The COAL supports and facilitates the reuse and sharing of joint community architectures, as well as rapid prototyping.[16]

The DRM is decomposed into the following operational activities, taken from (COAL v2.0, 2007):[17]

- Provide command and control decision support (COM.1.1)
    - Orient (COM.1.1.1)
- Conduct ISR/maintain battle-space awareness (Observe, COM.1.2)
    - Perform Shared Joint Intelligence Preparation of the Battlespace (COM.1.2.1)
    - Depict Location and Activities of all HVTs in Adversary Model (COM.1.2.1.4.4.4.4)
    - Develop and maintain shared awareness of the situation (COM.1.2.3)
- Understand the situation (COM.1.3)
    - Recognize threats (COM.1.3.1)
    - Assess Located Targets (COM.1.3.2)

[15] The Department of Defense Architecture Framework (DoDAF) serves as the overarching, comprehensive framework and conceptual model enabling the development of architectures to facilitate the ability of Department of Defense (DoD) managers at all levels to make key decisions more effectively through organized information sharing across the Department, Joint Capability Areas (JCAs), Mission, Component, and Program boundaries.

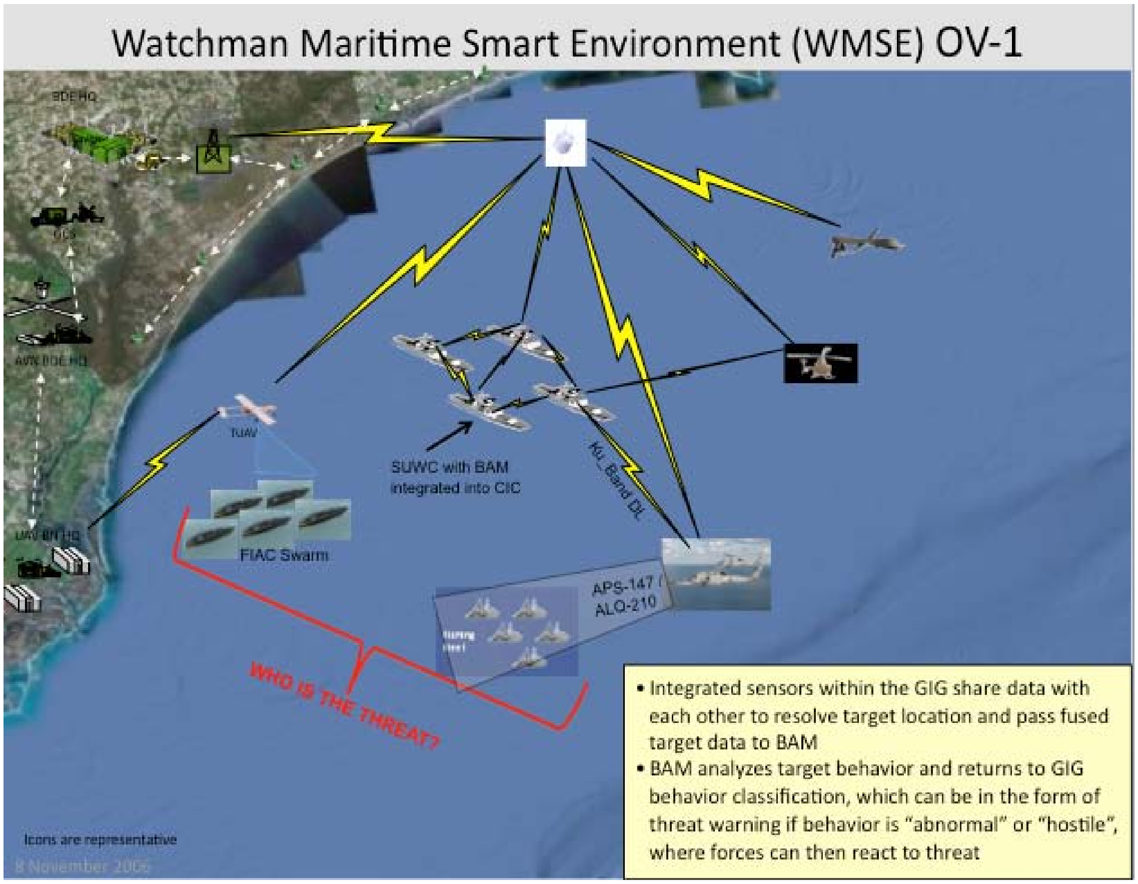
[16] FORCEnet Status Today – Briefing to the Strike, Land Attack & Air Defense (SLAAD) Division, Space and Naval Warfare Systems Center, April 29, 2004.

[17] Department of Defense (September 2007), "Common Operational Activity List COAL v2.0."

Once all operational activities have been identified, the functions necessary to achieve the mission will be identified and documented.

8. Mission Execution

Executing the mission will consist of completing certain tasks that can be traced back to their respective operational activities, described in the system architecture (Chapter IV). The High Level Operational Concept Graphic (OV-1) (Figure 1) shows a notional environment for the MSE system. Possible missions relating to this setting could be intercepting a single surface vessel that has come within the observation area, or several surface vessels that have broken away from within the fishing fleet and are heading toward the SAG.


Watchman Maritime Smart Environment (WMSE) OV-1

- Integrated sensors within the GIG share data with each other to resolve target location and pass fused target data to BAM
- BAM analyzes target behavior and returns to GIG behavior classification, which can be in the form of threat warning if behavior is "abnormal" or "hostile", where forces can then react to threat

Figure 1. MSE OV-1

In any of these situations, the MSE system will respond by completing the tasks it was programmed to enact when a threat is identified. The tasks taken from the Navy Tactical Task List (NTTL) 3.0[18] and the Universal Joint Task List (UJTL),[19] which will be identified for the DRM, are:

- Process and exploit collected information and intelligence
- Produce intelligence
- Provide Indications and Warning (I&W) of Threats
- Acquire, process, communicate information, and maintain status
- Analyze and assess situation
- Disseminate Tactical Warning Information and Attack Assessment

The CONOPS for these missions and their requisite tasks—being accomplished by the Behavior Analysis operational activity via a Behavior Analysis Module (BAM) algorithm (discussed in Chapter V)—must, by necessity, encompass more than just the BAM itself.[20] The BAM must be understood to operate in a larger context of a force protection schema, the Maritime Smart Environment (Figure 1), which consists of sensors, high-value units, communication, C^2 systems, and a network in which all of these members communicate and operate.

This section has described the purpose and scope of a DRM, as well as explained the specific DRM for this research, providing a context for the MSE, system architecture, and the Proof of Concept System that is described in later chapters.

[18] Department of the Navy (August 1, 2004), "Navy Tactical Task List (NTTL)," NTTL 3.0 (Draft).

[19] Department of the Navy (April 2007), "Joint and Naval Capability Terminology Lists (CMCL)."

[20] Rachel E. Goshorn, Deborah E. Goshorn, Joshua L. Goshorn, and Lawrence A. Goshorn, "Behavior Modeling for Detection, Identification, Prediction, and Reaction (DIPR) in AI Systems Solutions" Handbook of Ambient Intelligence and Smart Environments, Springer Handbook, http://www.springerlink.com/content/ n812r0064785g764/ (accessed April 18, 2010).

The next section discusses the approach that was pursued in the research for this thesis, including the process that was followed, how data was gathered, and the methods used in conducting the research.

C. THESIS RESEARCH APPROACH

This section discusses the approach that was taken in the conduct of the research for this thesis, first introducing the systems engineering process that was followed, and then introducing the development and documentation of the proof of concept system that was the focal point of the research.

1. Systems Engineering (SE) Process

The research for understanding different alternative solutions and finally arriving at the Proof of Concept System (POCS) as a model to address the CONOPS has progressed through a thorough SE process. As depicted in Figure 2, the Systems Engineering "Vee" model[21] was applied and followed, with some situational modifications. The first step in developing the POCS was to understand SE fundamentals and how to apply them to the design and development of a real, functional system. Beginning with requirements analysis, stemming from a needs analysis, and then leading to requirements definition and refinement, the concept for the POCS, from various alternative system solutions, began to take shape. From there, the system architecture was developed via a process of functional analysis and allocation, which then led to preliminary functional and process architectures, facilitating the initial design. Upon refinement of the design, specifications were created to align with requirements, and then the hardware installation and software coding processes could begin. As each hardware component was installed, connected, and tested, the various software modules were coded and tested independently. Subsequently, the hardware and software components were integrated incrementally, testing connections, interfaces, and increased functionality. As the errors from this incremental testing were corrected, the main

[21] Clifford Whitcomb, "Introduction to Systems Engineering" Lecture slides, Naval Postgraduate School, 2008.

modules were integrated, and then final full-up system testing and validation began. A more detailed description of the specific actions accomplished in each step of the SE process for each of the various project iterations is discussed in Chapter III.

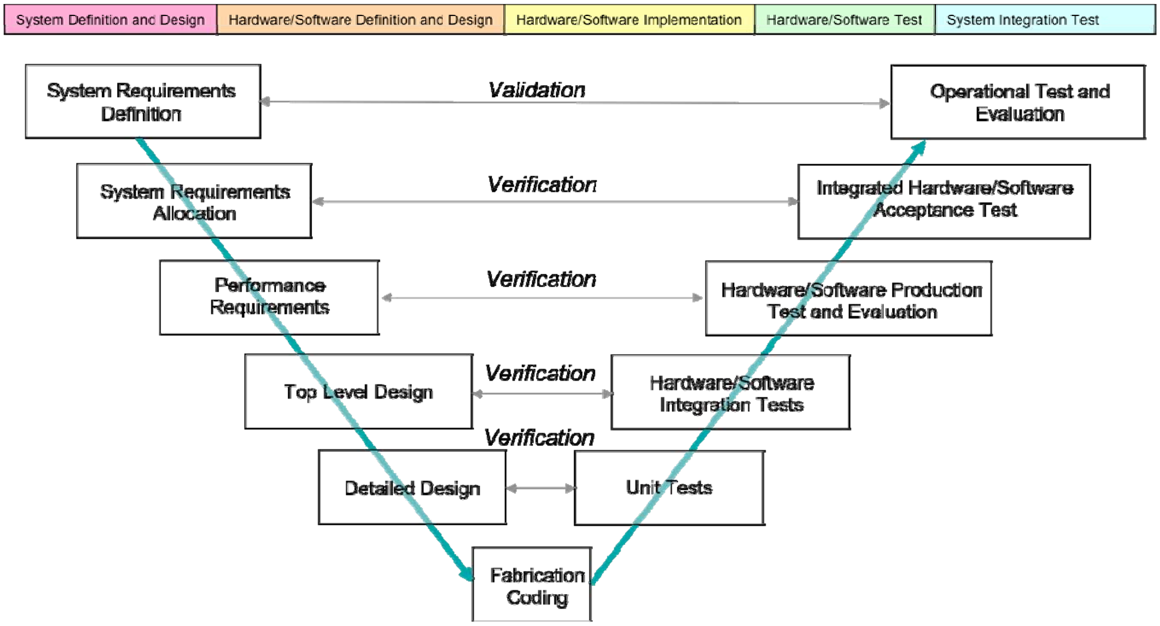

System Definition and Design
Hardware/Software Definition and Design
Hardware/Software Implementation
Hardware/Software Test
System Integration Test
System Requirements Definition
Validation
Operational Test and Evaluation
System Requirements Allocation
Verification
Integrated Hardware/Software Acceptance Test
Performance Requirements
Verification
Hardware/Software Production Test and Evaluation
Top Level Design
Verification
Hardware/Software Integration Tests
Verification
Detailed Design
Unit Tests
Fabrication Coding

Figure 2. Systems Engineering "Vee" Model (After Whitcomb)

After the full system was tested and demonstrated, a series of design iterations were implemented, going through a scaled down, yet very similar SE process, as described in the previous paragraph. These iterations were termed “Engineering Change Orders” (ECOs) by the advising professor. As the ECOs were being developed, coincident coursework led to a more discretely defined architecture, which yielded greater design improvements and efficiencies in the ECO design and integration processes.

A significant ECO improvement was the implementation of an enhanced Behavior Analysis algorithm in conjunction with camera calibration, which led to the ability of the system to define and analyze more discrete behaviors with higher spatial resolution. Along with these improvements, came the addition of a Wireless Smart Sensor Network (WSSN) of autonomous robotic agents for reaction to an identified target. The POCS has been thoroughly documented, and an extensive System

Description Document (SDD) defining the system's functional and operational architecture has been created. This process also helped to define requirements metrics for the system, as discussed in the DRM and in Chapter IV.

2. Proof of Concept System (POCS) Development and Documentation

In order to understand how MDA may be enhanced through the use of a Smart Sensor Network in a "Smart Environment," a POCS was developed and incrementally improved during the course of study at NPS. This POCS implemented a network of distributed sensors and processing to demonstrate the functions of the four stages of generalized intelligence automation: detection, identification, prediction, and reaction (DIPR), discussed in Chapter III.[22] This system, dubbed "Watchman," has been utilized for various thesis research projects, but certain aspects have been optimized and focused upon to represent a Maritime Smart Environment (MSE), employing BAM algorithms, akin to the Operational View depicted in Figure 1.

In addition to designing and developing the POCS, a full system architecture was developed as a result of the functional analysis and allocation steps of the SE process, which shows the required functions, processes and operational activities. These elements were derived from the systems requirements and are mapped showing the relationships with each other, in addition to the notional components, data elements, and informational linkages for a generic, as well as specific system solution.

Other research work accomplished to date on this project includes hardware and software modeling and mapping; documentation of the full POCS, along with changes that have been incorporated; a full System Description Document, which fully describes the entire system architecture, components and relationships; and finally system integration and testing—from component and module level, all the way to system-level demonstration testing—has been completed and documented.

Additionally, a thorough literature research review of current methods in MDA has been accomplished. The review studies governing documents for MDA

[22] Goshorn et al., "Behavior Modeling."

requirements, current technology and methods of MDA, current and future technology and methods of artificial intelligence and automation, and finally the current and future technology and methods of Network-Centric Warfare.

Furthermore, this thesis has been briefed at the Maritime Defense and Security. Research Program (MDSRP) monthly meeting held at the Naval Postgraduate School. This briefing was met with great interest and enthusiasm by the program director and other research group members in attendance. So affirming was the response that the director requested the research to be featured in an article in the group's monthly newsletter, SITREP.[23] From this newsletter article, various sectors have expressed keen interest in the promise of this research, from industry partners, to government research organizations, an on to Department of Defense requirements and procurement offices.

D. THESIS OUTLINE

This section explains how the rest of the thesis is laid out, with a short description of each of the rest of the thesis chapters, what those chapters include, and/or intend to accomplish for the reader.

1. The Current Methods of MDA

Chapter II broadly explains the contemporary types of technologies and systems for supporting MDA, with representative examples in order to provide the operational context in which MDA may be enhanced through the research explained in the succeeding chapters. It also presents some of the new "cutting edge" research areas in the field of automation in MDA. Furthermore, it explains the benefits of having an MSE-type system to enhance the current systems' execution of MDA.

[23] The NPS Maritime Defense and Security Research. Program Newsletter 43, February 2010, available at http://www.google.com/url?sa=t&source=web&ct=res&cd=1&ved=0CBIQFjAA&url=http%3A%2F%2Fwww.hsdl.org%2F%3Fview%26doc%3D119186%26coll%3Dpublic&ei=B4D9S_DrJoreNZ7Und4H&usg=AFQjCNF6TZQnqM8HqokieLYistNLwP1bpw (accessed May 4, 2010).

2. Systems Engineering Approach Overview

Chapter III further discusses what SE approach was utilized and how the SE process led to the development of the Watchman Maritime Smart Environment (WMSE) POCS. The chapter also explains how behavior analysis algorithms function in this system. The chapter additionally describes the steps of the SE process, as they pertained to each phase of development of the POCS. The details of progression through these steps will serve as a potential process model for the eventual integration of a scaled operational MSE system.

3. MSE System Architecture

Chapter IV presents the functional, process, and operational architectures of the WMSE POCS created as part of the functional analysis and allocation steps of the SE process. The chapter describes the architecture development process as a subset of the overall SE process for system alternative analysis and design development. The chapter further shows how the resultant architecture was ultimately applied in the design, building, coding, integration, and testing of the autonomous MSE POCS that represents an MDA "smart" environment.

4. Watchman MSE Proof of Concept System and Proposed Operational Implementation

Chapter V presents and describes the elements of the proof of concept/small scale laboratory system. This description includes the hardware and software design, layout (topology), and build, as well as improvements and upgrades made to the system after the initial development. Additionally, the chapter discusses the lessons learned from the system design, development, and integration processes, and what technology still needs to be refined for potential application and implementation of the system in an operational environment.

5. Summary and Conclusions

Chapter VI reinforces the need for a MSE system to enhance MDA, fulfilling certain capability gaps, and the importance of developing the technology to scale the POCS for operational use. The chapter also identifies and describes areas for further research, development, and testing.

This introductory chapter discussed the overview of the thesis, a description of the Design Reference Mission (DRM), the thesis research approach, and the outline of the thesis. The next chapter discusses the current methods employed in Maritime Domain Awareness.

II. CURRENT METHODS OF MDA

This chapter takes a brief look at some of the current means used to accomplish the MDA mission, from the sensors employed and their method of employment to the current net-centric operations (NCO) and data processing capabilities currently in the hands of our war fighters. Existing capabilities and new technologies are being examined to determine the most effective way to proceed in enhancing the ability to detect and track vessels and craft on the high seas. For example, large commercial vessels now carry a collision avoidance and harbor traffic control device called Automatic Identification System (AIS), which is comparable to Identification Friend or Foe (IFF) transponders fitted aboard military aircraft and commercial airliners.[24] The capabilities listed are not intended to be exhaustive, as there are myriad U.S., NATO, and other allied systems of many types for many different functions within the MDA sphere. Instead, this chapter is designed to give only a glimpse of the capability and variety of some of the representative systems in use for MDA today, in the realms of space, airborne, surface, subsurface and coastal.

Additionally, this chapter conducts a brief overview of some of the most recent Research and Development (R&D) initiatives in the world of MDA that are attempting to employ automation to accomplish mission tasks and goals.

A. CURRENT SENSORS AND CAPABILITIES

This section presents the overview of the representative MDA sensors and capabilities in all the media where MDA is conducted: Space, Air, Surface, Subsurface, and Coastal.

1. Space Sensors and Capabilities

The United States and our allies have launched reconnaissance satellites of various types and varying capabilities to provide required intelligence information. This

[24] Global Security, available at http://www.globalsecurity.org/intell/systems/mda.htm (accessed April 2, 2010).

information includes the activity in the maritime domain. There are four major types of reconnaissance satellites. *Early-warning satellites* can detect enemy missile launches. *Nuclear-explosion detection satellites* are designed to detect and identify nuclear explosions in space. *Photo-surveillance satellites, or imagery satellites, provide* photographs of enemy military activities, e.g., the deployment of intercontinental ballistic missiles (ICBMs), or the movements of illicit cargoes. There are two subtypes of reconnaissance satellites. “Close-look” satellites provide high-resolution photographs that are returned to earth via a reentry capsule, whereas “area-survey” satellites provide lower-resolution photographs that are transmitted to earth digitally via radio. Later satellites have combined these two functions. Other satellites use radar and thermal imaging technology to provide images of enemy activity when there is cloud cover or it is dark. *Electronic-reconnaissance (ferret) satellites* pick up and record radio and radar transmissions while passing over a foreign country or seagoing vessel.[25]

Other initiatives are attempting to enhance the capabilities of space sensors and communications to be more readily accessible to operational commanders in an effort to meet mission requirements, including MDA. The signature effort among these is Operationally Responsive Space (ORS). ORS has been broadly defined in DoD as assured space power focused on timely satisfaction of Joint Force Commanders’ (JFCs’) needs. ORS is thus considered a subset of space activities designed to fulfill JFCs’ needs, such as MDA surveillance, communication, and data exchange, while also maintaining the ability to address other users’ needs, ultimately improving the responsiveness of space capabilities to meet national security requirements. The ORS initiative is intended to create opportunities for integration and operational efficiencies needed to ensure affordable access to the space-based capabilities that are critical to fulfilling the full range

[25] “Reconnaissance Satellite,” The Columbia Encyclopedia, Sixth Edition. 2008. Encyclopedia.com. http://www.encyclopedia.com (accessed April 3, 2010).

of U.S. diplomatic, information, military, and economic needs, a specific goal of National Security Presidential Directive-49 (NSPD-49) on National Space Policy, dated August 31, 2006.[26]

Challenges in global political-military affairs are placing ever-increasing demands on the way the U.S. Armed Forces use space capabilities to achieve national security objectives, such as MDA. NSPD-49 reaffirms the United States' commitment to certain key principles that guide the conduct of space activities. Adhering to these principles will require the implementation of certain courses of action to achieve MDA goals and objectives. This will require additional global situational awareness, as well as adaptability to current and emerging threats, while acquiring the ability to advance the total suite of space capabilities to deal with these threats in new and innovative ways.[27]

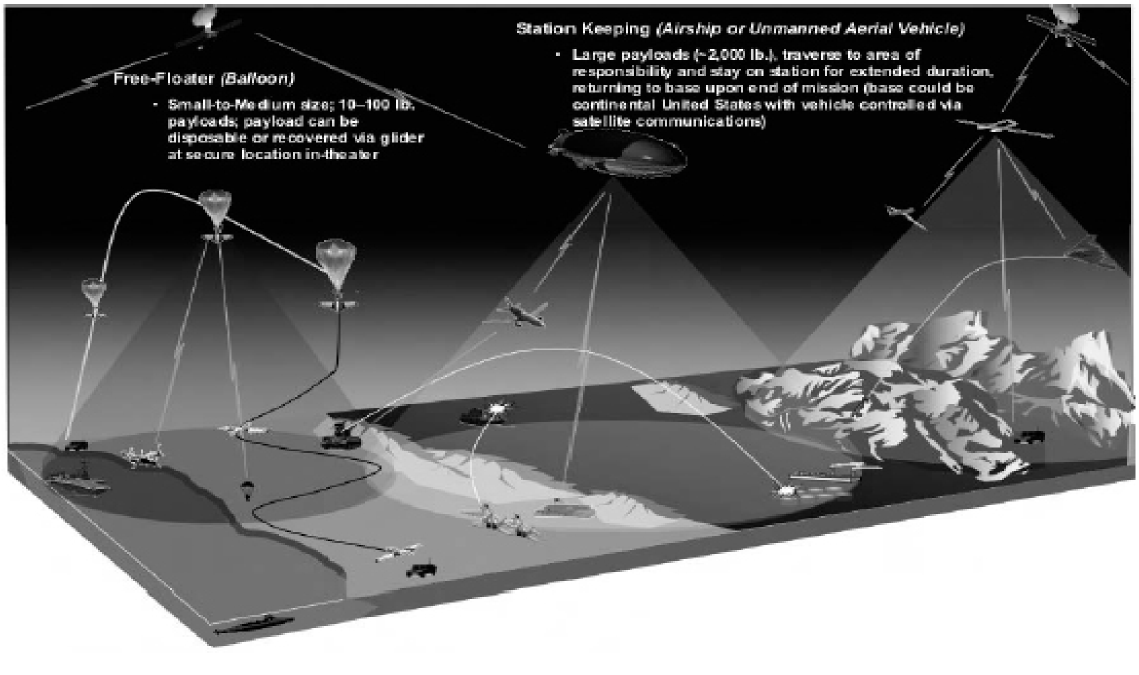

Free-Floater (Balloon)
• Small-to-Medium size; 10–100 lb. payloads; payload can be disposable or recovered via glider at secure location in-theater
Station Keeping (Airship or Unmanned Aerial Vehicle)
• Large payloads (~2,000 lb.), traverse to area of responsibility and stay on station for extended duration, returning to base upon end of mission (base could be continental United States with vehicle controlled via satellite communications)

Figure 3. Operationally responsive space: view of near-space architecture (From Doggrell)

[26] Plan for Operationally Responsive Space, A Report to Congressional Defense Committees, National Security Space Office (NSSO), April 17, 2007, 2, http://www.airpower.maxwell.af.mil/airchronicles/apj/apj06/sum06/doggrell.html (accessed April 8, 2010).

[27] Ibid.

One approach to addressing ORS for MDA and other mission needs entails not even going to space. This is based on the fact that terrestrial-based systems or aircraft, whether they be manned or unmanned, can meet many "space-type" needs. The Air Force has identified the realm above typical operational altitudes for aircraft and below the orbital regime (i.e., "space"), generally between 65,000 and 325,000 feet, as "near space," as shown in Figure 3. This very high altitude distinctively favors the operation of ISR, battlespace situational awareness, and communications assets, which are all vital to the execution of MDA. Unfortunately, near space has not been used extensively for military operations to date due to this operational environment's technical challenges (e.g., distance, environmental extremes). However, new advances in areas such as materials, solar collection, and power-storage technology can hopefully offer the opportunity to exploit this regime for MDA -applicable missions.[28]

2. Airborne Sensors and Capabilities

There are numerous airborne platforms with a varied collection of extensive capabilities that perform a vast array of detection, tracking, fusion, identification, classification, and strike functions in the maritime domain. Here are listed just a few prime examples of the current capability and flexibility in terms of broad-range to close-in-range sensing and engagement platforms.

The RQ-4A "Global Hawk" is an unmanned aerial vehicle (UAV) used by the United States Air Force and Navy as surveillance aircraft. The U.S. Navy had two RQ-4A air vehicles delivered in 2005, and in April 2008, the U.S. Navy selected the RQ-4N marinized variant of the Global Hawk RQ-4B Block 20 for the broad-area maritime surveillance (BAMS) unmanned aircraft system (UAS) requirement. The Global Hawk can carry out reconnaissance missions in all types of operations, in all weather and terrain conditions. The 14,000nm range and 42-hour endurance of the air vehicle, combined with satellite and line-of-sight communication links to ground or maritime forces, permits

[28] Les Doggrell, Operationally Responsive Space – A Vision for the Future of Military Space, Air & Space Power Journal, Summer 2006, available at http://www.airpower.maxwell.af.mil/airchronicles/apj/apj06/sum06/doggrell.html (accessed April 9, 2010.)

worldwide operation of the system. High-resolution sensors include visible and electro-optical / infrared (EO/IR) systems and synthetic aperture radar (SAR). A 10-inch reflecting telescope provides a common optics platform for EO and IR sensors. The EO/IR sensors operate in the 0.4 to 0.8 micron visible waveband and the 3.6 to 5 micron infrared band. In spot collection mode, the coverage is approximately 1,900 spots per day, with a spot size of 2km² to a geological accuracy of 20m circular error of probability (CEP). In wide-area search mode, the sensors can cover a swath that is 10km wide with total coverage at 40,000nm² a day. The SAR and ground moving target indicator (GMTI) operates in the X-band with a 600 MHz bandwidth, at 3.5kW peak power. The system can obtain images with 3-ft resolution in the wide-area search mode and 1-ft resolution in spot mode.[29]

The Navy's new MH-60R Multi-mission helicopter is a follow-on upgrade to and eventual replacement of the current Anti Submarine Warfare (ASW) and Anti-Surface Warfare (ASUW) helicopters, the SH-60B/F "Seahawk" series. This upgrade represents a significant avionics modification to the SH-60 series aircraft enhancing ASW, ASUW, surveillance and identification (ID), as well as power projection missions, supporting the operational requirements of full-dimension MDA. The upgrade includes the "dipping" Airborne Low Frequency Sonar (ALFS) and greatly increases sonobuoy and acoustic signal processing using a Commercial of the Shelf (COTS) Enhanced Modular Signal Processor. In addition, the aircraft employs an entirely new Multi-Mode Radar (MMR), including Inverse Synthetic Aperture Radar (ISAR) detection and imaging and periscope detection modes. The MMR also includes an embedded data fusion system, which permits standoff classification of hostile threats, as to specifically what type, class, and capability. Also, the Electronic Support Measures (ESM)[30] system upgrade now exceeds the capabilities found on U.S. Navy surface ships and includes a fully automated and

[29] Net Resources International, http://www.airforce-technology.com/projects/global/ (accessed April 2, 2010.

[30] Electronic Support Measures (ESM) is a passive type of Electronic Warfare (EW) that exploits Signal Intelligence (SIGINT), made up of Electronic Intelligence (ELINT) and Communications Intelligence (COMINT), in order to locate, identify, and track enemy contacts via their emitted electromagnetic radiation signals, http://www.radartutorial.eu/16.eccm/ja05.en.html (accessed April 14, 2010).

integrated Radar Warning System (RWS) for self-protection.[31] The AN/ALQ-210 ESM system on the MH-60R is a multiple-bandwidth phase, frequency and amplitude-measuring receiver. Incorporating digital receiver technology, the system utilizes wide instantaneous bandwidth to provide enhanced probability of intercept as well as frequency-agile emitter acquisition. Featuring these technologies, it has been observed that this ESM system locates both moving and stationary targets to an order of magnitude far exceeding the capability of classic Direction-Finding (DF) equipment. Enhanced threat identification and mode determination is accomplished via operationally proven algorithms and digital parametric accuracy. The AN/ALQ-210 provides autonomous processing to reduce operator workload, combined with manual intervention to allow for tailoring of parameters.[32]Additionally, the aircraft employs a Forward Looking Infrared (FLIR) sensor, with laser designator/rangefinder, providing the capability to launch Hellfire missiles. The MH-60R weapons suite also includes MK-48 and MK-50 analog and digital ASW torpedoes.[33]

The P-3C “Orion” long-range ASW aircraft is a four-engine turboprop ASW and maritime surveillance aircraft. Originally designed as a land-based, long-range, ASW patrol aircraft, the P-3C's mission evolved in the late 1990s and early 21st century to include surveillance of the entire battlespace, either at sea or over land or in between, in the littorals. The P-3C has advanced submarine detection sensors such as directional frequency and ranging (DIFAR) sonobuoys and magnetic anomaly detection (MAD) equipment. Additionally, the P-3C has improved ASUW capabilities via the Antisurface-warfare Improvement Program (AIP), which incorporates enhancements in its ASUW suite, such as the ASX-6 Multi-Mode Imaging System (MMIS), over-the-horizon targeting (OTH-T), command, control, communications and intelligence (C4I), and improved survivability. The avionics system is integrated by a general-purpose digital

[31] Global Security, http://www.globalsecurity.org/military/systems/aircraft/sh-60r.htm (accessed April 5, 2010).

[32] Jane's Avionics, Airborne electronic warfare (EW) systems, United States, February 3, 2010, http://www8.janes.com/JDIC/JDET (accessed April 5, 2010).

[33] Global Security, http://www.globalsecurity.org/military/systems/aircraft/sh-60r.htm (accessed April 5, 2010).

computer that supports all of the tactical displays, monitors and can automatically launch ordnance, as well as provide flight information to the pilots. In addition, the system coordinates navigation information and receives sensor data inputs for tactical display and storage. The P-3C can carry a mixed payload of both ASW and ASUW weapons internally and on wing pylons.[34]

3. Surface Sensors and Capabilities

As one would expect, the U.S. Navy employs many capabilities for MDA mission execution in the primary medium of the maritime environment, the surface of the seas. These capabilities are numerous, and deployed aboard the entire range of surface ships, from large combatants—such as aircraft carriers, cruisers, destroyers, frigates, amphibious assault ships, and the new multi-mission littoral combat ship—all the way down to the smaller mine-sweepers, patrol craft, and even auxiliaries, such as supply and hospital ships. Rather than discussing the specific platforms, and enumerating the myriad capabilities resident on each, which would be quite lengthy, we will instead look at some of the more common sensors utilized across the spectrum of the U.S. surface fleet, and discuss their capabilities in prosecuting threats and other targets of interest in the maritime domain.

The AN/SPY-1 multifunction fixed phased-array radar is the equipment that forms the central part of the U.S. Navy's Airborne Early warning/Ground environment Integration Segment (AEGIS) fleet air defense system. As applied to the AEGIS cruisers and destroyers, the system's phase-scanned arrays are mounted in pairs on the vessel's fore and aft deckhouse, in such a way as to provide 360° radar coverage. The radar’s output is measured in megawatts and it operates in the S-band, with the transmitter serving several parallel channels simultaneously. Multiple radiating elements within each array are digitally controlled by computer to facilitate the production and steering of multiple radar beams for target search, detection, and tracking, as well as the generation of target track data for associated ownship missile target illumination and command

[34] U.S. Navy Fact File, P-3C Orion, http://www.navy.mil/navydata/fact_display.asp?cid=1100&tid=1400&ct=1 (accessed 5 April 5, 2010).

guidance. The AN/SPY-1 had been developed and deployed in a number of variants, upgrading capability and functionality, along with the AEGIS system.[35]

The AN/SPS-55 solid-state I-band surface search and navigation radar is another widely deployed system. It is designed for employment on surface ships of destroyer size or above, and has been installed on almost every capital ship in the U.S. Navy. It is operationally used for: navigation and pilotage; the detection of small surface targets ranging from less than 50m up to the radar horizon; tracking of low-flying aircraft and helicopters; detection of submarines at snorkel and periscope depth. The system's lightweight, low profile antenna minimizes installation space requirements. The antenna consists of two selectable back-to-back, end-fed, slotted arrays, one with circular and the other with linear-horizontal polarization. With a horizontal beamwidth of 1.5º and beam "squint" compensation, the system can optimize bearing accuracy over the entire operating frequency range. Vertical beamwidth is 20º. The SPS-55's transceiver subsystem is housed below-deck in a single cabinet and is capable of operating at any selected frequency in the band from 9.05 to 10 GHz, with two pulse-widths (1 and 0.12 µs) and variable sector radiation provided. The minimum peak transmitter output is 130 kW. To permit remote operation of the transmitter/receiver and scanner subsystems, a separate control unit is provided, as the SPS-55 set does not normally include its own display.[36]

A newer system, similar to the SPS-55, is the AN/SPS-67(V) surface search radar. This is a solid-state G-band system that was originally designed to replace the AN/SPS-10 surface search radar, the antenna that was initially used with the new equipment. Aside from the antenna, the AN/SPS-67 consists of an antenna controller and an antenna safety switch, a transceiver, a video processor, and a radar control unit. System performance was enhanced for better navigation and improved resolution of small targets at short ranges by the addition of a very narrow pulse mode (0.1 µs). In open sea, long and medium pulse modes (1 and 0.25 µs) are used for detection of long- and medium-

[35] Jane's Radar and Electronic Warfare Systems, March 30, 2010.

[36] Ibid., November 10, 2000.

range targets. A digital video clutter suppressor and an interference suppressor further improve performance. *Jane's* sources report that AN/SPS-67(V) series radars have been installed aboard the following warships and classes of warship: the aircraft carrier *Enterprise*; Nimitz class aircraft carriers; Arleigh Burke (Flts I, II and IIA)-class destroyers; Austin-class amphibious transport docks; Tarawa and Wasp-class amphibious assault ships and Mercy-class hospital ships.[37]

It is extremely challenging to detect and, especially, identify potential maritime threats or illicit behavior in total darkness or in adverse weather conditions. Thermal imaging cameras and other Forward-Looking Infrared (FLIR) systems have been developed to help maritime domain operators to meet the demands of identifying contacts of interest at night and in other low-light situations. Thermal imaging cameras can be integrated with radars and other sensors in the maritime environment in a so-called "slew-to-cue" mode, which allows operators to automatically aim the thermal imaging system at a radar track for automatic target acquisition and rapid identification.[38]

The SAFIRE (AN/AAQ-22)/ SeaFLIR (AN/KAX-1/2)/SeaFLIR III/SEA Star SAFIRE III family of systems are surface platform-based infrared sensors for passive detection and tracking of surface targets. The four-axis stabilized sensor package in the naval version SAFIRE is 38.1 cm in diameter, 44.6 cm high and weighs 38.1 kg. The sensor package uses a 4×4 cadmium-mercury-telluride detector array with a split-Stirling, closed-cycle, cooling to cover the 8–12 μm spectral band with ×1.9 and ×10.5 magnification. Naval versions also include de-icing. The below-deck control unit fits into a 19 in (48.3 cm) rack and weighs 12.15 kg, with buttons and a joystick with various sizes of monochrome display available. The system uses standard video formats and digital interfaces to ESM and navigation systems. Optional extras include a digital auto-tracking feature, video recording, a MIL STD 1553B data bus interface with ship navigation controls and displays tactical symbology.

[37] Jane's Radar and Electronic Warfare Systems, August 14, 2009.

[38] Thermal Imaging Cameras for Border Security and Coastal Surveillance, FLIR Systems, 2010, http://www.flir.com/cvs/eurasia/en/content/?id=9652 (accessed April 14, 2010).

The SeaFLIR (AN/KAX-1/2) features a smaller, lighter director with a Indium Antimonide Focal Plane Array detector (256×256) covering the 3–5 µm wave band, having a resolution of 1.2 to 0.12 mrad with 10×1 continuous zoom. The sensor fits into a 23 cm gimbaled mount. Three versions of SeaFLIR are available; long range color CCD TV sensor, with an 8.46 mm, 811 × 608 CCD with 470 line output and 48 × 2.7° field-of-view, low-light monochrome CCD TV sensor with a 12.7 mm, 768 × 576 CCD with 570 line output to provide a 29 × 2.9° field-of-view, and laser range-finder, which features a 1.5 µm eye-safe range-finder with a range resolution of +/-5m.[39]

The AN/SLQ-32(V) is the U.S. Navy's standard surface ship radar detection, analysis and jamming system. It is a wide-open (in angle and frequency) ESM system that covers radar threats within the U.S. Band 3 frequency range. It also provides cues and controls for shipboard Electronic Counter Measures (ECM) decoy launchers and, if fitted, an active radar jammer. The system is comprised of two multi-beam antenna/receiver assemblies, each of which is equipped with a 180° semi-omni sense antenna (for instantaneous frequency measurement) and two 90° Direction-Finding (DF) lens arrays. It also includes processing and display equipment. A special purpose digital processing subsystem incorporating DF/frequency correlation and digital tracking elements correlates coarse frequency and amplitude data, which, together with signal time-of-arrival, is used to create a pulse descriptor word. These emitters are then catalogued and stored in the emitter file memory of the digital tracker. If three or more pulses meeting a specific signature are received within a programmable time span, the system's AN/UYK-19 central computer is informed that a new emitter has been detected. This detection triggers a command for the digital tracker to provide additional stored pulses for full-scale analysis. From this analysis, the threat signal's parameters - pulse repetition frequency, scan type, scan rate and frequency—are established in order to allow the system to identify the emitter against a digital emitter library. Once the

[39] Jane's Naval Weapon Systems, September 16, 2009, http://www8.janes.com/JDIC/JDET (accessed April 1, 2010).

identification process is complete, the system then generates appropriate alert signals to system operators and commands for the ECM systems.[40]

4. Subsurface Sensors and Capabilities

Whereas the primary detection medium above the surface of the water is exploitation of the electromagnetic spectrum, beneath the surface, targets must be detected, tracked, classified, and engaged almost exclusively by the use of acoustic sensors. Some of the most effective employment of these types of sensors is through those platforms that operate in the same environment as the subsurface threat, which is the U.S. submarine force. Here we will look at some of those subsurface capabilities, which are used for MDA in the underwater domain.

The Virginia Class attack submarine is an advanced, stealthy nuclear-powered submarine designed for deep ocean ASW and littoral operations. The Virginia Class employs the AN/BQQ-10(V4) sonar processing system for its acoustic sensor suite, which includes bow-mounted active and passive arrays, side-mounted wide aperture passive array, high-frequency active arrays on keel and fin, TB-16 towed array and the TB-29A thin line towed array. In place of traditional optical periscopes, the submarinesare fitted with Kollmorgen AN/BVS-1 photonic mast. The sensors mounted on this mast include LLTV (low-light TV), thermal imager and laser rangefinder. To facilitate simultaneous communication at super high frequency (SHF) and extremely high frequency (EHF), submarine high data rate (sub HDR) multi-band satellite communications systems are also mounted on the mast. The Boeing long-term mine reconnaissance system (LMRS) will be deployed on the Virginia Class, which includes two 6m autonomous unmanned underwater vehicles (UUV), an 18m robotic recovery arm along with support electronics.

The command and control systems module (CCSM) will integrate all of the vessel's systems, to include sensors, countermeasures, navigation, weapon control, and

[40] Jane's Avionics, Airborne electronic warfare (EW) systems, United States, December 8, 2008, http://www8.janes.com/JDIC/JDET (accessed April 5, 2010).

will be based on open system architecture (OSA) with Q-70 common display consoles. Weapons control will be provided with the AN/BYG-1 combat control system (CCS), a derivative of the CCS MK 2 combat system, which is also being fitted to the Australian Collins Class submarines.[41]

The other cutting edge subsurface platform is the Seawolf attack submarine, which was conceived as a faster, better-armed replacement for the Los Angeles Class attack submarines. However, the Seawolf's intended use as a counter to Soviet submarines was deemed not worth the cost, given the end of the Cold War. This change of emphasis to littoral operations led to preferring the smaller and cheaper Virginia Class, with only three Seawolves constructed.

The Seawolf submarines are being upgraded with the LAN/BQQ-10(V4) sonar processing system to manage the submarine's sonar suite that includes: BQQ 5D bow-mounted active / passive arrays and wide aperture passive flank arrays; TB-16 surveillance and TB-29A thin-line towed array; BQS 24 active sonar for close range detection. The CCS is a Lockheed Martin BSY-2 with a network of approximately 70 68030 Motorola processors. This system will eventually be replaced by the AN/BYG-1 combat system, with weapons control managed by the MK 2 fire control system.[42]

As mentioned previously, there are several air launched or deployed acoustic sensors, such as sonobouys and dipping sonar, for detecting and tracking subsurface targets. In addition to these assets, there are also several surface-based acoustic sensors, such as hull-mounted and towed array sonar systems. A few examples of those systems are described below.

The AN/SQS-53 is the most advanced surface ship ASW sonar in the U.S. Navy. It is a high-power, long-range system, used both actively and passively to prosecute submarine contacts. The SQS-53B, located at the bow of the ship, detects, localizes,

[41] NSSN Virginia Class Attack Submarine, USA, http://www.naval-technology.com/projects/nssn/ (accessed April 7, 2010).

[42] SSN Seawolf Class Attack Submarine, USA, http://www.naval-technology.com/projects/nssn/ (accessed April 7, 2010).

classifies, and tracks multiple subsurface contacts. With improved signal processing, this sensor was the first sonar in the Navy linked directly to digital computers for rapid and accurate target processing. The system also has the capabilities for underwater communications, countermeasures against acoustic underwater weapons, and certain oceanographic recording.

The AN/SQS-53C version has both active and passive modes for ASW weapons control and guidance. The AN/SQS-53C performs direct path ASW search, detection, localization, and tracking from the hull mounted transducer array. The AN/SQS-53C's computer-aided detection and automatic contact management supports the system's high source level, fully stabilized beams, and wide convergence zone annuli. The AN/SQS-53C provides apparent range, bearing, and true bearing of contacts in active mode and true bearing of contacts in passive mode. The AN/SQS-53C is a digital system with stable performance, fail-safe features, and automated performance monitoring/fault isolation.[43]

Another surface acoustic system is the AN/SQR-19 Tactical Towed Array SONAR (TACTAS), which provides very long-range passive detection of subsurface contacts. TACTAS consists of a long cable full of hydrophones that is towed about a mile behind the surface platform. It is towed far enough behind to prevent ship self-noise from interfering with target noise signatures. The system provides the ability to detect, classify, and track a large number of submarine contacts at increased ranges. TACTAS is a component sensor of the AN/SQQ-89(V)6 ASW Combat System, providing significant improvements in passive detection and localization, with 360-degree coverage at tactical ship speeds.[44]

TACTAS is also employed upon Tactical Auxiliary General Ocean Surveillance (T-AGOS) ships, which have the singular mission of gathering underwater acoustical

[43] Global Security, http://www.globalsecurity.org/military/systems/ship/systems/an-sqs-53.htm (accessed April 7, 2010).

[44] Military Analysis Network, http://www.fas.org/man/dod-101/sys/ship/weaps/an-sqr-19.htm (accessed April 7, 2010).

data, as part of a system called the Surveillance Towed Array Sonar System, or SURTASS. The T-AGOS ships operate to support the anti-submarine warfare mission of the Maritime Component Commanders.

T-AGOS ships are designed to tow several types of underwater listening devices to collect acoustical data. They also carry advanced electronic equipment for processing and transmitting data for evaluation via satellite to shore stations. Other services provided by these civilian contractor-manned ships are oceanographic and hydrographic surveys and acoustic research. SURTASS patrols are typically 60 to 90 days in duration.[45]

SURTASS is an element of the Integrated Undersea Surveillance System (IUSS), being developed and deployed for more mobile, tactical capability in the early 1980s, providing long-range detection and cuing for tactical weapons platforms against both diesel and nuclear submarines. The fixed component of the IUSS is the Sound Surveillance System (SOSUS). SOSUS consists of hydrophone arrays on the sea bottom connected by undersea communication cables to facilities ashore.[46] With SOSUS arrays being placed in a standby status after the end of the Cold War, where data was available but arrays were not continuously monitored, SURTASS must provide the undersea surveillance necessary to support regional conflicts and Sea Lines of Communication (SLOC) protection.

The SURTASS Block Upgrade expands the capabilities of the towed array collection and analysis system on T-AGOS ships. The upgrade improved sensitivity and signal processing and use of a reduced diameter hydrophone array with advancements against quiet threats. The upgrade added the Reduced Diameter Array (RDA) and a COTS processing system that provides improved detection capability and target bearing accuracy and better spectrum analysis. Communications upgrades provide additional UHF SATCOM voice and data connectivity between T-AGOS and tactical platforms.

[45] Navy Fact File, http://usmilitary.about.com/library/milinfo/navyfacts/blsurveillanceships.htm (accessed April 7, 2010).

[46] National Oceanographic and Atmospheric Association (NOAA) VENTS Web site, http://www.pmel.noaa.gov/vents/acoustics/sosus.html (accessed April 16, 2010).

The SURTASS Low Frequency Active (LFA) system upgrade is the active addition to this towed array. LFA is a long-range active sensor designed to detect even quieter threats in the future, including a large source array for active transmissions and an array as a separate receiver. LFA also includes an active signal processing and display system, an active transmit array and handling system, power amplification and control systems and an environmental analysis system to the SURTASS Upgrade. [47]

5. Coastal Sensors and Capabilities

Given that the United States is a maritime nation with hundreds and hundreds of miles of coastline, coastal sensors for the monitoring of the maritime domain nearest the Homeland provide a vital MDA capability. The implementation of measures for maritime security has brought greater attention to the need for coastal surveillance. The current implementation of policy on maritime security is also routinely addressing operational responses to humanitarian and environmental issues, which drives the demand for surveillance data. The apparent lack of information on activities within the coastal zone supports the need to develop Coastal Domain Awareness by establishing the COP of all potential threats within the coastal zone.[48] This section describes a few of the representative systems that are utilized to meet the coastal COP development mission.

Cardion E/F-band (2 to 4 GHz) automatic coastal surveillance radar has been developed and produced as a transportable 10cm band coastal defense radar system which can rapidly be deployed in remote sites for coastline monitoring and early warning. The system includes a Secondary Surveillance Radar (SSR) system and was designed for use at unattended sites with automatic reporting of data and control signals. An extensive self-monitoring system has been incorporated into the design, allowing for remote deployment and operation. The system automatically reports alerts are transmitted upon detections or abnormal conditions. The main radar antenna is a double curvature reflector

[47] Military Analysis Network, http://www.fas.org/irp/program/collect/surtass.htm (accessed April 7, 2010).

[48] Mark Womersley, Protecting Coastal Communities through Civil Maritime Surveillance, http://www.gisdevelopment.net/application/nrm/coastal/mnm/art_001pf.htm (accessed April 7, 2010).

design with integrated primary and secondary feed assemblies. A secondary omnidirectional antenna is mounted on top of the main antenna and used for sidelobe suppression in the SSR. Since the radar is typically employed in a coverage augmentation role, the vertical beam has been specifically adapted to provide high detection probability of small surface targets and low-altitude aircraft[49]

The LCR 2020 is a frequency agile, 2-Dimensional, coastal surveillance radar system designed for simultaneous detection of surface and air targets. The system is described as exhibiting “superior” surface detection capabilities, coupled with all-weather performance. To accomplish these capabilities, the smart sensor features adaptive processing, with a robust land/sea clutter rejection, as well as incorporating a suite of jamming detection and anti-jam electronic counter-countermeasures (ECCM) capabilities. Other features include autonomous operation and a network-capable architecture.[50]

The Orion coastal air defense radar is a pulse-Doppler F-band (3 to 4 GHz) coastal surface and low-altitude air radar system. The Orion system is designed to provide surveillance coverage over designated land and sea areas through coastal surveillance, low-altitude air space surveillance and inshore coastal/harbor traffic monitoring. The coastal surveillance function can provide early warning and monitoring of all types of large surface vessels, in addition to a variety of small craft. The low-altitude air surveillance function allows the monitoring of low-flying aircraft and helicopters. Orion has been designed for remote operation, to be deployed at unattended sites with automated transmission of data and control signals. The system includes the F-band primary search radar and the D-band (1 to 2 GHz) SSR.

The Orion system can be deployed in either fixed-site, transportable, or mobile installation configurations. The fixed-site configuration provides persistent surveillance of potential threat approaches, with the antenna tower mounted with or without radome,

[49] Jane's Radar and Electronic Warfare Systems, November 10, 2000, http://www8.janes.com/JDIC/JDET (accessed April 2, 2010).

[50] Ibid., January 8, 2009.

and can break down for transport by land, sea or air. In the transportable configuration, the system may operate from semi-permanent sites, with a telescopic, transportable tower that extends the horizon surveillance coverage. The entire system is housed in a military equipment shelter, which can also be transported by land, sea, or air. In the mobile configuration, the system is housed and rapidly deployed in a tactical tracked or wheeled vehicle, with a vehicle-mounted antenna. This setup is also highly transportable.[51]

B. DATA FUSION

This section discusses the field of data fusion and how it applies to MDA by first presenting the theory of data fusion, then explaining data fusion in practice, and concluding with examples of data fusion in operational use.

1. Data Fusion Theory

Combining various data from sensors (both similar and dissimilar sensors) is sensor fusion and combining that data together with current-state and a priori knowledge can be considered information fusion.[52] Faceli, Carvalho, and Rezende define sensor fusion as "the combination of information from different sensors to capture data of the environment whose obtaining is beyond the capacity of each sensor individually, mainly when reliability and precision are considered." Sensor fusion, in pulling together and combining attributes from many sources, can provide to users new information that none of the sensors alone could supply.[53] Figure 4 demonstrates a basic architecture of one approach to sensor fusion, showing how the many different types of data, each with their individual informational attributes, are fused together to create a synergistic product of combined characteristics. These combined characteristics define entirely different, additional, and arguably more useful information to the user, than the constituent parts could provide separately.

[51] Jane's Radar and Electronic Warfare Systems, November November 10, 2000.

[52] R. P. Bonasso, "What AI can do for battle management," *AI Magazine* 9, 1988, 77–83.

[53] David C. Schafer, A Systems Engineering Survey of Artificial Intelligence And Smart Sensor Networks In A Network-Centric Environment, Thesis, Naval Postgraduate School, September 2009.

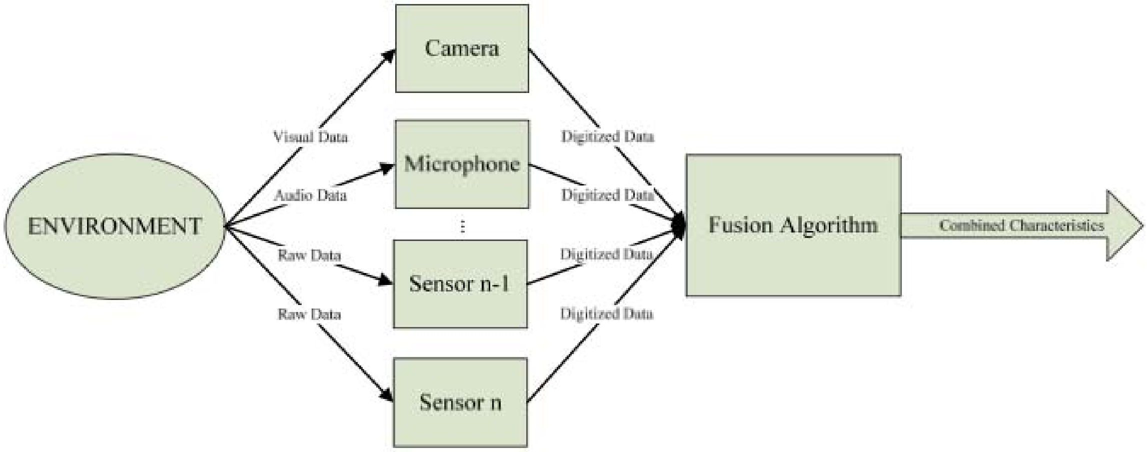


Figure 4. Basic Architecture of Sensor Fusion. (After Schafer)

Four different types of sensor fusion are specified below:[54]

a. Complementary Fusion

Different types of sensors provide different (complementary) views of the environment (e.g., video and audio data provide more accurate data on a target).

b. Competitive Fusion

The goal of competitive fusion is to provide redundant information about the area within the environment observed. More than one sensor observes the same feature of the environment (e.g., two cameras observing the same space, but obtain differing figure of merit, which can be compared for quality of data).

c. Cooperative Fusion

Cooperative fusion is the combination of data from independent sensors in order to obtain information that could not be obtained by any of the sensors by

[54] K. Faceli, A. C. P. L. F. de Carvalho, and S. O. Rezende, "Combining intelligent techniques for sensor fusion," *Proceedings of the 9th International Conference on Neural Information Processing, 2002 (ICONIP '02),* vol. 4, 1998–2002, IEEE Press, New York, Pub. No. 981-04-7524-1, 2002.

themselves (e.g., a radar and an ESM system on either the same or separate platforms working in conjunction with one another, sharing target attributes to ascertain a target's identity).

d. Independent Fusion

Unrelated sensors provide information to a common storage location in a common data structure (e.g., satellite data and AIS data on a commercial tanker provided independently to a database for eventual fusion and pattern learning).

2. Data Fusion in Practice

Under the construct of Net-Centric Operations (NCO), "Horizontal Fusion" is considered the catalyst for net-centric transformation in DoD. Horizontal Fusion is one of the pillars of the Department's NCW/NCO transformation effort, which includes vital infrastructure elements, such as Global Information Grid Bandwidth Expansion (GIG-BE), Joint Tactical Radio System (JTRS), Wideband Satellite Communications (SATCOM), Net-Centric Enterprise Services (NCES), and Information Assurance (IA).The goal of Horizontal Fusion is to provide real-time situational awareness (SA) through the use of battlespace, "sense-making" tools, collaboration among communities-of-interest, and the ever-elusive critical intelligence information sharing.

The term "horizontal" refers to the ability to provide connection and communication across traditionally stove-piped peer-to-peer organizations; and "fusion" refers to the products and processes that facilitate sensor and information "melding." It is envisioned that users will have the ability to gain access to the additional, augmented information that data fusion provides, across the battlespace, through "smart-pull" systems and applications for full-spectrum information sharing. DoD's Office of Force Transformation describes this process by the verbs task, post, process, and use (TPPU). Through TPPU, users have rapid access to information through "smart-pull" methods. A critical aspect to the success of TPPU, is the interoperable infrastructures that are required within the DoD and across all associated intelligence-gathering organizations. Geographically dispersed commanders and forces will have the power to act as one interconnected team through the sharing of the COP. Another boon to information sharing is the real-time collaboration among users to share best practices and contribute to the available body of knowledge, regardless of their respective communities of interest.

This subsection discussed the concept of data fusion and presented different types, as well as an operational methodology of data fusion; the next subsection provides examples of systems that are currently used to accomplish data fusion in support of MDA.

3. Operational Data Fusion Examples

The Global Command and Control System-Maritime (GCCS-M) is the designated Command and Control (C^2) system for the Navy, as well as the naval implementation of the national GCCS. GCCS-M supports numerous operational and intelligence missions for commanders at every level, in all naval environments (afloat and ashore) for joint, coalition, and allied forces. GCCS-M, implemented across all naval command centers, is able to exchange data among approximately 20,000 users for near real-time SA critical to operational and tactical analysis and decision making. GCCS-M meets joint and service requirements for a single, integrated, scalable C^2 system that receives, displays, correlates, fuses, and maintains geo-located track information on friendly, hostile, and neutral land, sea, and air forces. It then integrates this track data with available intelligence and environmental information. GCCS-M fully supports the developing concepts for Network-Centric Operations through the fusing and integrating track, intelligence and other available relevant information for the operators.[55]

Trusted Information Service (TIS) program is a compilation of the Multi-Level Security (MLS) capabilities of the Navy's complementary Ocean Surveillance Information System (OSIS) and Radiant Mercury systems. This system will facilitate expansion and extension of the Commander's ability to automatically exchange essential fused intelligence and operational information with all networked forces.

[55] Faceli, de Carvalho, and Rezende, "Combining intelligent techniques for sensor fusion."

The OSIS Evolutionary Development (OED) system is the DoD's premier C4I processing and dissemination system, forming the basis of the automated information infrastructure, supporting the COP at U.S. and allied Joint Intelligence Centers (JICs). OED receives, processes, fuses, and disseminates timely all-source surveillance information on fixed and mobile targets of interest, across the maritime domain, within an MLS environment. OED facilitates collaboration in multiple domains, via viewing, monitoring, and analyzing multiple views of the battle space at all security classification levels. The effective communication and fusion subsystems of OED provide extremely rapid delivery of battlespace informational products in support of the Unified Combatant Commanders, Joint Task Force commanders, individual units, and allies. MLS is envisioned as serving as the core fusion and collaboration technology upon which future Navy networks and databases across multiple classification levels can be effectively joined to allow appropriately cleared operators access to fused data from a single workstation.

Radiant Mercury (RM) provides the vital operations security functions that automatically sanitize, transliterate, and downgrade classified, formatted information to users at lower classification levels. RM helps ensure critical Indications and Warning (I&W) intelligence is provided quickly to operational decision makers at the various classification levels. RM is currently fielded in maritime platforms bridging data transfer between SCI GCCS-M and GENSER GCCS-M. Radiant Mercury Imagery Guard (RMIG) combines a digital signature process with RM allowing the networked transfer of imagery between security domains, which can then be fused with other data sources.[56]

The next section discusses how all of these capabilities and processes are brought together and exploited for power capitalization and projection in the Information Age.

C. NETWORK-CENTRIC WARFARE

Network-Centric Warfare, or “Net-Centric Warfare” (NCW) is a theory of warfare that intends to exploit the combat power that can be harnessed and wielded in the

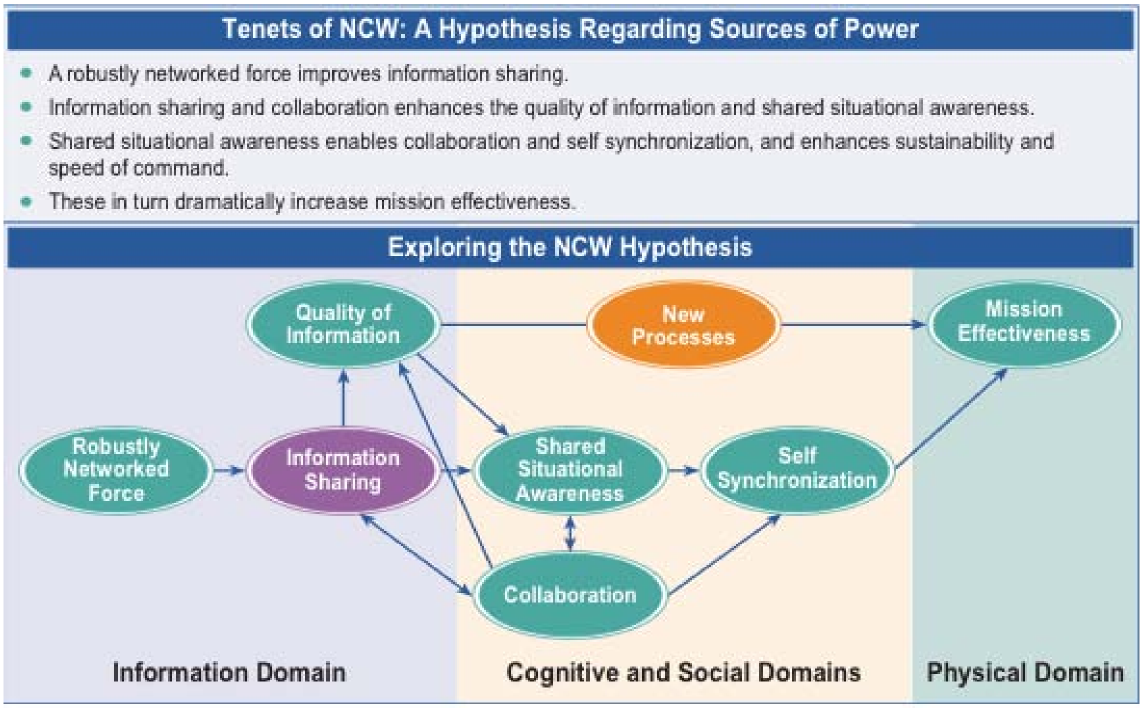
[56] Faceli, de Carvalho, and Rezende, "Combining intelligent techniques for sensor fusion."

effective control of information. This control stems from being able to link military entities in a network, so that they form a connected, information-sharing war fighting enterprise. Geographically dispersed entities within the battlespace must be able to share and manage a high level of information that can be exploited via self-synchronization and other network-centric operations to achieve commanders' intent. NCW supports the requirement to convert superiority of information position to action, enhancing the speed of command. NCW can be applied across the spectrum of force size, terrain, location, and mission. Additionally, NCW is key to bridging the tactical, operational, and strategic levels of war.[57]

To successfully implement the emerging theory of war and NCW capabilities, under the construct of Network Centric Operations, the four domains of warfare—physical, information, cognitive, and social—must be understood, as well as the intersections, or areas of overlap, between the domains. The four basic tenets of NCW, as shown in Figure 5, constitute a hypothesis regarding NCW as a source of power.[58]

[57] D. S. Alberts, J. J. Garstka, F. P. Stein (2000), Network Centric Warfare: Developing and Leveraging Information Superiority, CCRP Publ., 2nd Edition (Revised). Aug 1999, Second Print February 2000.

[58] United States Dept. of Defense (DoD). Office of Force Transformation (OFT), *The Implementation of Network-Centric Warfare,* U.S. Government Printing Office, Washington, D.C., 2005.


Tenets of NCW: A Hypothesis Regarding Sources of Power

- A robustly networked force improves information sharing.
- Information sharing and collaboration enhances the quality of information and shared situational awareness.
- Shared situational awareness enables collaboration and self synchronization, and enhances sustainability and speed of command.
- These in turn dramatically increase mission effectiveness.

Exploring the NCW Hypothesis

Figure 5. The Tenets and Domains of NCW (From DoD OFT)

D. NETWORK-CENTRIC OPERATIONS

The implementation of Network Centric Warfare is through Network-Centric or "Net-Centric" Operations (NCO). To conduct NCO, war fighters must apply the tenets and principles of NCW. A force that can properly conduct NCO is "more adaptive, ready to respond to uncertainty in the very dynamic environment of the future at all levels of warfare and across the range of military operations."[59] As joint forces become more effectively networked—sharing a common operating picture, and therefore have shared situational awareness—they can communicate more efficiently and the operational effectiveness of joint force increases.

> A networked Joint Force is able to maintain a more accurate presentation of the battlespace built on the ability to integrate intelligence, surveillance, and reconnaissance, information and total asset visibility. This integrated picture allows the [Joint Force Commander] to better employ the right

[59] United States Dept. of Defense. Office of Force Transformation, *The Implementation of Network-Centric Warfare*

> capabilities, at the right place and at the right time. Fully networked forces are better able to conduct distributed operations.[60]

Data pulled from Operation Enduring Freedom (OEF) (2001–2002) and Operation Iraqi Freedom (OIF) (2003) in Afghanistan and Iraq, respectively, provide case studies in the effectiveness in conducting NCO.

During OEF, the military saw a shift from platform-centric to network-centric operations as weapons platforms were successfully networked with sensor platforms. Ground units were networked to each other, as well as with aircraft and Unmanned Aerial Vehicles (UAVs), which provided commanders a near-real time battlefield situational awareness, as well as an unparalleled ability to effectively engage the enemy.

Improving upon the success gained in OEF, OIF saw not only the fusion of sensors within a network, but also a fusion of the war fighter within an integrated networked, joint force. Brigadier General Dennis Moran said: “The ability to move intelligence rapidly from the sensor to either an analytical decision maker or directly to the shooter was the best that we have ever seen... We validated the concept of network-centric warfare.”[61]

The experiences of both OEF and OIF have shown that increasing the capability and the quality of networks in all domains, will result in better information sharing, enhanced rapidity of communications, and therefore an increased speed of command. Even though these experiences demonstrate the validity of NCO in a joint environment, one must ask in the context of this thesis, “How is NCO applied specifically in a maritime environment? In the next section, the U.S. Navy’s implementation of network-centric operations is discussed.

1. FORCENet

The U.S. Navy’s FORCENet concept is “the operational construct and architectural framework for naval warfare in the information age, integrating warriors,

[60] Department of Defense, Joint Operations Concepts, November 2003, 16.

[61] Ibid.

sensors, command and control, platforms, and weapons into a networked, distributed combat force[62]." Admiral Vern Clark, Chief of Naval Operations, and General Michael Hagee, Commandant of the Marine Corps, approved the FORCEnet Functional Concept in February 2005, in order to provide the shared direction, guiding principles, and future capabilities for the Navy and Marine Corps Net-Centric force development, as well as network command and control efforts. This concept, derived directly from the Naval Operating Concept (2015-2020) for Joint Operations, was developed under the Joint Capabilities Integration and Development System (JCIDS) process and fully supports the Navy's vision of Naval Power 21, as well as the combined strategies of Sea Power 21 and Marine Corps Strategy 21. The concept is meant to capitalize on the power of networking decision makers, which will increase war fighting capabilities and improve overall combat effectiveness and mission accomplishment through a fully integrated and interactive fighting force.[63]

According to the Navy's FORCENet Website, "the objective of FORCEnet is to provide commanders the means to make better, timelier decisions than they currently can and to see to the effective execution of those decisions." It explains an underlying premise called the *network effect* that makes FORCEnet such a powerful concept. This *network effect* "causes the value of a product or service in a network to increase exponentially as the number of those using it increases." The concept assumes that most headquarters entities are already well connected, so the goal of FORCEnet is to further connect the extremities of the force, otherwise known as "disadvantaged users,"[64] such as individuals, weapons, sensors, platforms, munitions, shipments, end items, parts, and so on. FORCEnet is intended to extend visibility and empowerment to these disadvantaged

[62] "Sea Power 21, Projecting Decisive Joint Capabilities" Admiral Vern Clark, U.S. Navy Proceedings, October 2002, http://www.navy.mil/navydata/cno/proceedings.html (accessed April 15, 2010).

[63] Global Security, http://www.globalsecurity.org/military/systems/ship/systems/forcenet.htm (accessed March 25, 2010).

[64] R. E. Goshorn, "Findings for network-centric systems engineering education," presented at the Military Communications (MILCOM) Conference, San Diego, November 2008.

users. Senior Navy leadership believes that the concept's greatest leap forward will be to achieve future command and control through "maximum decentralization."[65]

The Functional Concept identifies 15 capabilities that are necessary to implement the FORCEnet Concept:

1. Provide robust, reliable communication to all nodes, based on the varying information requirements and capabilities of those nodes.

2. Provide reliable, accurate and timely location, identity and status information on all friendly forces, units, activities and entities/individuals.

3. Provide reliable, accurate and timely location, identification, tracking and engagement information on environmental, neutral and hostile elements, activities, events, sites, platforms, and individuals.

4. Store, catalogue and retrieve all information produced by any node on the network in a comprehensive, standard repository so that the information is readily accessible to all nodes and compatible with the forms required by any nodes, within security restrictions.

5. Process, sort, analyze, evaluate, and synthesize large amounts of disparate information while still providing direct access to raw data as required.

6. Provide each decision maker the ability to depict situational information in a tailorable, user-defined, shareable, primarily visual representation.

7. Provide distributed groups of decision makers the ability to cooperate in the performance of common command and control activities by means of a collaborative work environment.

8. Automate certain lower-order command and control sub-processes and to use intelligent agents and automated decision aids to assist people in performing higher-order sub-processes, such as gaining situational awareness and devising concepts of operations.

9. Provide information assurance.

[65] "FORCEnet, A Functional Concept for the 21st Century" Admiral Vern Clark, U.S. Navy General Michael Hagee, U.S. Marine Corps, February 7, 2005.

10. Function in multiple security domains and multiple security levels within a domain and manage access dynamically.

11. Interoperate with command and control systems of very different type and level of sophistication.

12. Allow individual nodes to function while temporarily disconnected from the network.

13. Automatically and adaptively monitor and manage the functioning of the command and control system to ensure effective and efficient operation and to diagnose problems and make repairs as needed.

14. Incorporate new capabilities into the system quickly without causing undue disruption to the performance of the system.

15. Provide decision makers the ability to make and implement good decisions quickly under conditions of uncertainty, friction, time, pressure, and other stresses.[66]

Additionally, the concept states that in order to fully realize these capabilities, developmental efforts are required across six dimensions:

1. Physical – The various platforms, weapons, sensors and other entities on the operating end of FORCEnet.

2. Information technology – The communications and network infrastructure through which these entities interact.

3. Data – The common structure and protocols for information handling.

4. Cognitive – Human judgment and decision making and the human-computer interfaces that support them.

5. Organizational – The new force structures and working relationships that will be made possible by FORCEnet.

6. Operating – The emergent methods and concepts by which forces and other organizations will accomplish their missions due to the capabilities provided by FORCEnet.[67]

[66] “FORCEnet, A Functional Concept for the 21st Century.”

[67] Ibid.

The understanding of the power of decentralization in command and control is absolutely critical in the implementation of distributed processing, which is, in turn, a key architectural tenet of ABA integration. The concepts of distributed processing and integration architecture for system design is explored more fully in Chapters III and IV, respectively. However, suffice it to say at this point that the U.S. Navy's vision for implementation of NCO in the maritime domain under FORCENet is ideal for the concept of integration of Automated Behavior Analysis capability within its Net-Centric framework.

Furthermore, this thesis will establish how the concept of ABA integration squarely meets the FORCENet implementation capabilities 6, 8, 11, 12, 14, and 15 listed, as well as spans all six dimensions of development for those capabilities.

E. CURRENT AUTOMATION RESEARCH AND DEVELOPMENT (R&D) INITIATIVES

This section highlights some of the most recent and relevant R&D efforts in the field of autonomous data fusion and analysis in the maritime domain, to include some efforts in automated behavior or pattern analysis. Although these initiatives all show differing levels of technological maturity and focus on several operational concept dilemmas in the maritime domain, they do all share the attributes of networked systems built for the purpose of automating analysis of vast amounts of maritime data to reduce operator workload and enhance threat response.

The systems engineering involved in the development of these types of systems is the subject of a section in Chapter III, so it will not be discussed here. Also in that section, as well as in Chapter V, the key differences between the automation processes and the systems engineering approaches of these systems and those of the Watchman Maritime Smart Environment Proof of Concept System are thoroughly addressed. What is helpful to note here, is that the research conducted under this thesis is neither redundant nor mutually exclusive of any of the projects described herein. On the contrary, this thesis provides a unique approach and perspective that can both benefit from, as well as add to the insights that these efforts have garnered in the course of research on the topic.

It will be highly recommended that further research stemming from this thesis should include collaboration among these promising endeavors.

1. Predictive Analysis for Naval Deployment Activities (PANDA)

The PANDA project intends to advance technologies and develop the architecture for a system that has the ability to alert watch standers to anomalous ship behavior in real time, allowing them to identify potentially dangerous behavior and react in a timely manner. The current CONOPS for achieving situational awareness in the maritime domain requires close and continuous monitoring of those ships or cargoes that have been previously flagged for concern (e.g., a suspected threat or illicit cargo) by some other intelligence gathering means. The goal of PANDA is to autonomously assess the behavior of all larger surface maritime vessels, to ascertain which vessels might be deviating from their normal, patterned behavior that may reveal hostile intent.

The vision of PANDA is to go beyond merely tracking objects to begin to perform autonomous motion-based pattern analysis on those objects' tracks and their correlated activities. The goals of autonomous motion-based pattern analysis are to:

- Learn normal patterns of behavior for objects of interest.
- Leverage those patterns to perform motion-based anomaly and change detection.
- Provide anticipatory situation awareness by alerting our military forces to emerging threats and changes in our enemies' operating patterns.

To date, the ability to learn motion-based patterns has been hampered by the inability to perform persistent surveillance and conduct long-duration tracking on contacts of interest. Current enhancements in long-duration surveillance tracking in the maritime domain now make this motion-based pattern learning possible.

The challenges to accomplishing this vision are many. Activities that are required to further the vision and goal are:

- Research in Motion-Based Pattern Learning (learn motion-based activity models for individual, groups, and/or classes of vessels)
- Prediction and Activity Monitoring (compare the incoming and emerging vessel tracks/behaviors with the existing vessel patterns and determine if the current behavior is consistent with the learned pattern)
- Adaptive Context Modeling (learn, maintain, and efficiently store "business process" models related to variations in surface maritime ship behavior as well as suspicious behaviors)
- Anomaly Processing and Presentation (compare anomalies generated to the anomaly database to determine if the anomaly is an emerging threat and present the results for review/resolution).

The project scope is segmented into phases, focusing on the various aspects of the required activities: Phase I – learning and detection; Phase II – automation; Phase III – integration; Phase IV – technology scaling and transition. Throughout the phases of the program, technology developers, integrators, and testers will work closely with data providers and system end-users to make certain that the system delivered provides the essential information and is fully integrated with existing systems. This project will eventually transition the technologies and systems to various partners and customers throughout the development process, to ensure that the developed system ultimately meets user needs.[68]

2. Maritime Automated Scene Understanding (ASU) System

An Automated Scene Understanding (ASU) system is intended, much as the PANDA project describes, to relieve watch standers of critical yet mind-numbing surveillance and detection duties that free them to focus on threat assessment and response. Just as was discussed in the WMSE CONOPS in terms of military assets,

[68] DARPA information processing Techniques Office (IPTO)/Programs/Predictive Analysis for Naval Deployment Activities (PANDA), http://www.darpa.mil/ipto/programs/panda/panda_goals.asp (accessed April 8, 2010).

government agencies and commercial ports authorities are collecting vast amounts of data through the deployment of video, radar, and other sensor assets to monitor the maritime domain. As it has already been stated, this incredible amount of data cannot be effectively monitored and analyzed by existing manpower.

SeeCoast is an ASU system designed to provide additional MDA capability. It has been developed by BAE Systems under the Department of Homeland Security (DHS) Science and Technology (S&T) Program. Built from a suite of components developed by BAE over the past 12 years, the system is designed to provide a real-time understanding of the maritime scene with minimal operator oversight. SeeCoast is designed to detect and track vessels (including estimating vessel lengths), fuse tracks, learn normal behavior patterns of surface vessels, and generate alerts for anomalous patterns. The data the system is designed to provide can ostensibly enhance maritime situational awareness by providing comprehensive vessel tracking and alerting operators to unusual vessel behaviors. The autonomous nature of the system will hopefully allow operators to focus on threat assessment and response rather than surveillance and detection. The system is currently being evaluated at Joint Harbor Operation Center (JHOC) Hampton Roads, Virginia.

SeeCoast works by processing many diverse real-time tracks from various sensors such as radar, AIS, and cameras. Disparate tracks data is fused to establish a single track for each vessel, as well as to increase track confidence and accuracy. Video processing of an autonomously controlled network of pan-tilt-zoom camera data can detect, track, and estimate lengths of vessel. Camera calibration along with vessel waterline approximation technology enables the system to geographically locate tracks, as well as classify vessels by size. The situational understanding and alerting module learns normal vessel activity to create normalcy models by cataloguing and storing vessel features such as vessel size (class), unique ID (AIS), and time period. Vessel feature patterns are analyzed in real-time against the learned models to detect anomalous behavior and generate alerts. False alert rates are controlled via threshold and model parameter settings. The normalcy models can be automatically updated based on operator responses to alerts.[69]

3. Comprehensive Maritime Awareness (CMA)

The CMA Joint Capability Technology Demonstration (JCTD) was designed to rapidly assemble a maritime picture from multiple worldwide data sources through the use of automated data acquisition, fusion tools, anomaly detection capabilities, enhanced collaboration, and effective net-centric data management strategies. The JCTD's desired end state was to enhance command-level maritime awareness, facilitate operational decision making, and increase opportunities for international and interagency maritime security cooperation.[70]

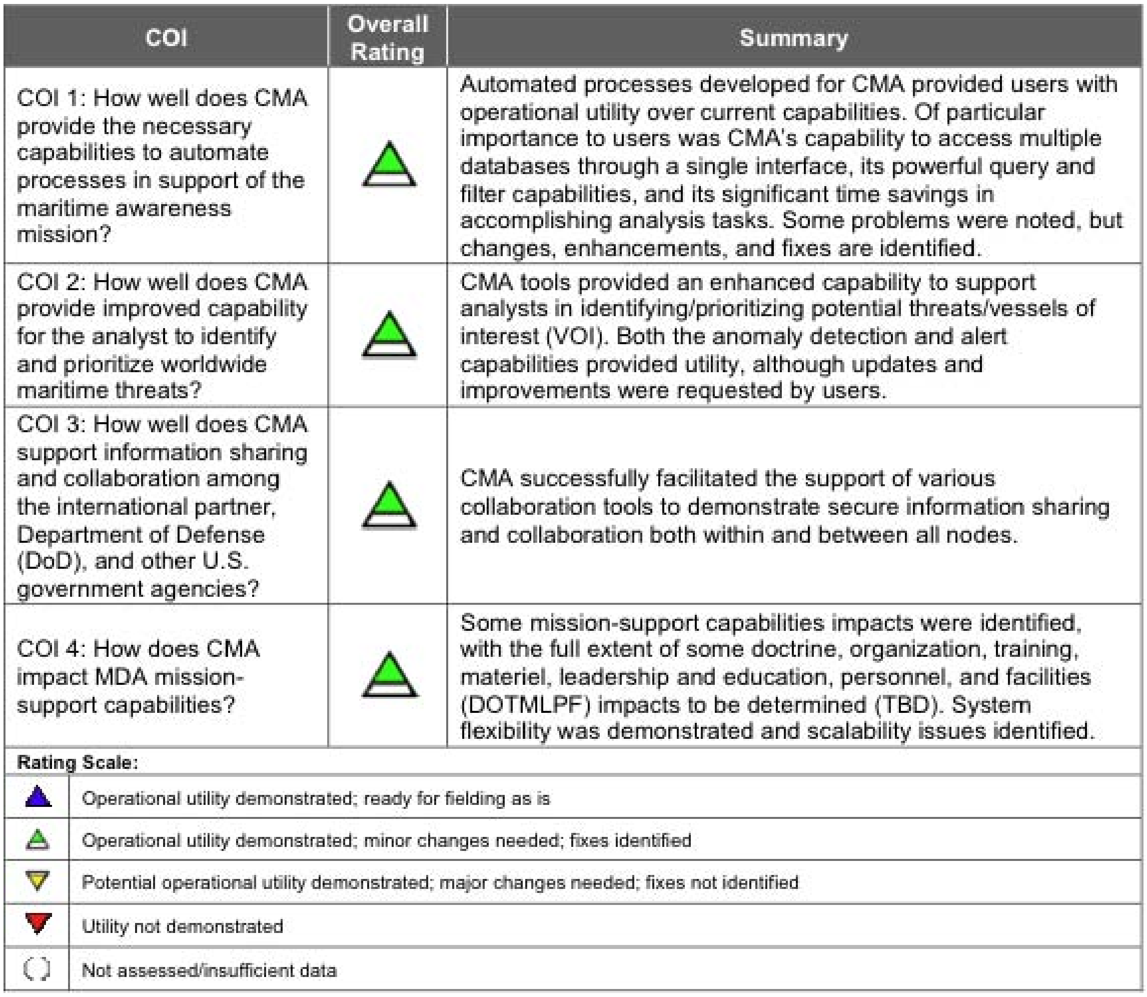
The CMA system was designed to pull together a comprehensive maritime picture from myriad data sources using various technologies to include: data acquisition, automated data fusion, vessel behavior anomaly detection, collaboration, and net-centric data management. Similar to this thesis research, the goal of CMA is to enhance maritime domain awareness via facilitating operational decision making. A unique aspect of CMA is that it also seeks to increase opportunities for international and interagency maritime security cooperation. The Office of the Secretary of Defense (OSD) authorized the CMA JCTD to swiftly field technologies to help reduce the potential maritime threat vulnerability of U.S. commercial shipping and ports. This section discusses the results of the last phase of the JCTD, which was an Operational Utility Assessment (OUA)[71] in which operators used and evaluated the latest development of CMA system capabilities.[72]

[69] Michael Seibert, Bradley J. Rhodes, Neil A. Bomberger, Patricia O. Beane, Jason J. Sroka, Wendy Kogel, William Kreamer, Chris Stauffer, Linda Kirschner, Edmond Chalom, Michael Bosse, and Robert Tillson, SeeCoast Port Surveillance, Proceedings of SPIE Vol. 6204: Photonics for Port and Harbor Security II Orlando, FL, April 18–19, 2006.

[70] CMA JCTD Management Plan (Revision A), February 20, 2007, 1–1.

[71] OUA: An evaluation of operational effectiveness and operational suitability made by an independent operational test activity, with user support as required, on other than production systems.

[72] CMA JCTD Operational Utility Assessment Final Report, May 2009, 6.



COI	Overall Rating	Summary
COI 1: How well does CMA provide the necessary capabilities to automate processes in support of the maritime awareness mission?	△	Automated processes developed for CMA provided users with operational utility over current capabilities. Of particular importance to users was CMA's capability to access multiple databases through a single interface, its powerful query and filter capabilities, and its significant time savings in accomplishing analysis tasks. Some problems were noted, but changes, enhancements, and fixes are identified.
COI 2: How well does CMA provide improved capability for the analyst to identify and prioritize worldwide maritime threats?	△	CMA tools provided an enhanced capability to support analysts in identifying/prioritizing potential threats/vessels of interest (VOI). Both the anomaly detection and alert capabilities provided utility, although updates and improvements were requested by users.
COI 3: How well does CMA support information sharing and collaboration among the international partner, Department of Defense (DoD), and other U.S. government agencies?	△	CMA successfully facilitated the support of various collaboration tools to demonstrate secure information sharing and collaboration both within and between all nodes.
COI 4: How does CMA impact MDA mission-support capabilities?	△	Some mission-support capabilities impacts were identified, with the full extent of some doctrine, organization, training, materiel, leadership and education, personnel, and facilities (DOTMLPF) impacts to be determined (TBD). System flexibility was demonstrated and scalability issues identified.

Rating Scale:	
▲	Operational utility demonstrated; ready for fielding as is
△	Operational utility demonstrated; minor changes needed; fixes identified
▽	Potential operational utility demonstrated; major changes needed; fixes not identified
▼	Utility not demonstrated
◌	Not assessed/insufficient data

Table 4. CMA JCTD OUA Overall COI Ratings[73]

Based on the OUA, CMA demonstrated an analytical capability enhancement to current MDA methods and tools. Table 4 shows the overall ratings for the critical operational issues (COIs) during OD4. The fused maritime information provided by CMA was considered a noticeable improvement over current capabilities. However, it was determined that operational use will be limited until a fully reliable 24/7 data update feature with sufficient bandwidth is implemented.[74]

[73] CMA JCTD Operational Utility Assessment Final Report, May 2009, 6.

[74] Ibid., 2.

CMA capabilities were noted as facilitating identification of high-interest maritime activity by the user and that the capability to identify vessels showing signs of anomalous behaviors is a potent instrument for analysts. Users recognized that continued development and integration of this capability would be necessary for it to be useful operationally. During the demonstrations, significant saving of time was realized through CMA's automated track development, querying, filtering, and display capabilities. Also, information sharing and collaboration among participants was successful.

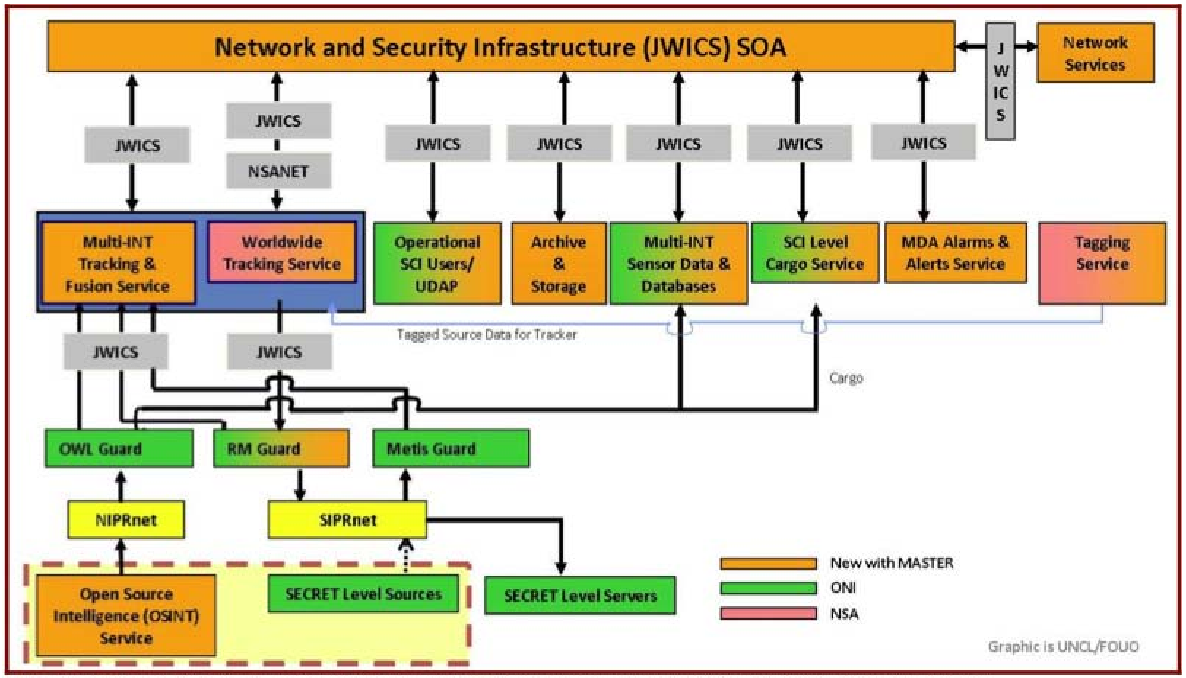
Based on the assessment results, the Joint Test and Assessment Activity (JTAA) recommended that CMA capability be fielded and sustained. The JTAA recognized that full capability is yet to be realized. Several developmental efforts must be undertaken to make certain the system provides consistent, reliable data and stable operations 24/7.[75]

4. Maritime Automated Super Track Enhanced Reporting (MASTER)

MASTER was another JCTD commissioned to rapidly field new technology, however, the vision of the MASTER JCTD is to improve MDA by drawing from multiple data sources from different security levels to support creation of a "Super Track" - meaning a track containing vessel data gained from multiple sources which has been fused together - integrated on a single, user-defined operational picture at the Top Secret/Sensitive Compartmented Information (TS/SCI) level on the Joint Worldwide Intelligence Communications System (JWICS). Preceding the development of MASTER, this disparate data was not readily available, if available at all, on a single display. MASTER has the ability to automate data fusion by using a metadata fusion engine to permit the operator to identify and prioritize maritime threats.[76]

[75] CMA JCTD Operational Utility Assessment Final Report, May 2009, 2, 3.

[76] Maritime Automated Super Track Enhanced Reporting (MASTER) Joint Capability Technology Demonstration (JCTD) Operational Demonstration 2 Operational Utility Assessment Report, April 2009, 5.


Network and Security Infrastructure (JWICS) SOA
JWICS
Network Services
JWICS
JWICS
NSANET
JWICS
JWICS
JWICS
JWICS
JWICS
Multi-INT Tracking & Fusion Service
Worldwide Tracking Service
Operational SCI Users/ UDAP
Archive & Storage
Multi-INT Sensor Data & Databases
SCI Level Cargo Service
MDA Alarms & Alerts Service
Tagging Service
JWICS
JWICS
Tagged Source Data for Tracker
Cargo
OWL Guard
RM Guard
Metis Guard
NIPRnet
SIPRnet
Open Source Intelligence (OSINT) Service
SECRET Level Sources
SECRET Level Servers
New with MASTER
ONI
NSA
Graphic is UNCL/FOUO

INT = intelligence; NIPRnet = non-secure Internet protocol router network; NSA = National Security Agency; NSANET = National Security Agency Network; OWL = one-way link; RM = Radiant Mercury; SIPRnet = Secret Internet Protocol Router Network; UDAP = user-defined awareness picture

Figure 6. MASTER System View[77]

MASTER uses a flexible Service Oriented Architecture (SOA) as the base infrastructure for all MASTER services and activities. These include data collection, anomaly detection and alerting, support of operator threat assessment, search capabilities, and network data sharing. All of these capabilities were available, implemented, and assessed in OD2. MASTER's SOA directly enables aggregation, access, and handling of maritime data by external systems and operators. Analysts at the TS/SCI level on JWICS can access integrated sensor data from multiple database sources on MASTER due to the design of new hardware and software. Figure 6 provides a system-level view of MASTER.[78]

MASTER track data can transmitted to a SIPRnet site from JWICS via approved network classification security protocols. These protocols allow secret-level maritime

[77] MASTER JCTD Operational Demonstration 2 Operational Utility Assessment Report.

[78] Ibid.

data to be used by other MDA systems such as CMA. A significant Open Source Intelligence (OSINT) service on NIPRnet provides maritime data from several hundred active, open sources for inclusion into the MASTER tool suite. Other protocols are used to transmit Open/Sensitive but Unclassified data sources over NIPRnet up to the TS/SCI level on JWICS. The Sensitive but Unclassified data can then be passed to the MASTER database for fusion with JWICS. MASTER fuses these open data sources with TS/SCI-level data sources to enable operators to generate a more comprehensive MDA analysis picture.[79]

Chapter II took a brief look at some of the current means, in both sensors and platforms employed, to accomplish the MDA mission. Additionally, this chapter carried out a succinct overview of some of the most recent R&D initiatives in the area of MDA that are attempting to employ automation to accomplish mission tasks and goals. The next chapter provides an overview of the SE approach that was utilized and how the SE process led to the development of the Watchman Maritime Smart Environment (WMSE) proof of concept system (POCS). The chapter will further explain how behavior analysis algorithms function in this system. The chapter will additionally describe the steps of the SE process, as they pertained to each phase of development of the POCS.

[79] MASTER JCTD Operational Demonstration 2 Operational Utility Assessment Report.

III. SYSTEMS ENGINEERING APPROACH OVERVIEW

The Systems Engineering (SE) Approach Overview chapter discusses what SE approach was utilized and how the SE process led to the development of the Watchman Maritime Smart Environment (WMSE) proof of concept system (POCS). The chapter begins with a generalized view of an SE approach to Artificial Intelligence, in particular Behavior Analysis, which will serve to explain how behavior analysis algorithms function within this system. The chapter concludes by describing the steps of the SE process, as they pertained to each phase of development of the POCS.

A. THE SYSTEMS ENGINEERING OF BEHAVIOR ANALYSIS

This section looks at a particular approach to engineering Artificial Intelligence systems, known as the “AI Systems Solution Pyramid.” This look delves in detail into one of the pyramid’s levels of generalized intelligence automation fusing, the Detect, Identify, Predict, React, or “DIPR” software development construct.

1. The Artificial Intelligence Systems Pyramid

Artificial Intelligence (AI) systems have the capability to automate the means by which concepts are characterized and represented in order to provide a relevant logic mechanism for use in decision making.[80] As the maritime domain becomes more network-centric and flooded with sensors of all types, the plethora of data (required by both the end user/operators and decision-makers) that is produced must be ever more efficiently processed. It has become essentially implausible for human operators to adequately sift, evaluate, integrate and convey to other operators the appraisal of this incredible amount of data. AI systems have shown great potential in assisting operators

[80] G. Winstanley, "Artificial intelligence support for systems engineering," *IEE Colloquium on IT Support for Systems Engineers*, 6/1–6/5, IEEE Press, New York, Pub. No. 3844604, 1990.

with the increased reliability and efficiency of data collection and analysis. To harness this potential, network-centric systems engineers should have a firm grasp and working knowledge of AI systems concepts.[81]

One such concept is the innovative perspective on the SE of AI systems that forms the basis for the Watchman system, the "AI Systems Solution Pyramid," shown in Figure 7. This figure demonstrates a generalized view of an AI Systems Solution where the levels of the pyramid represent systems that comprise the entire solution. Behavior Analysis, as will be shown, forms an integral part of this system architecture perspective.

From the discrete mission requirements, the total system solution starts both from the "top-down" with one or more tailored applications and from the "bottom-up" with a system infrastructure. Those infrastructures can either be a currently operational, modified, or newly developed infrastructure designed and built as part of the system solution.

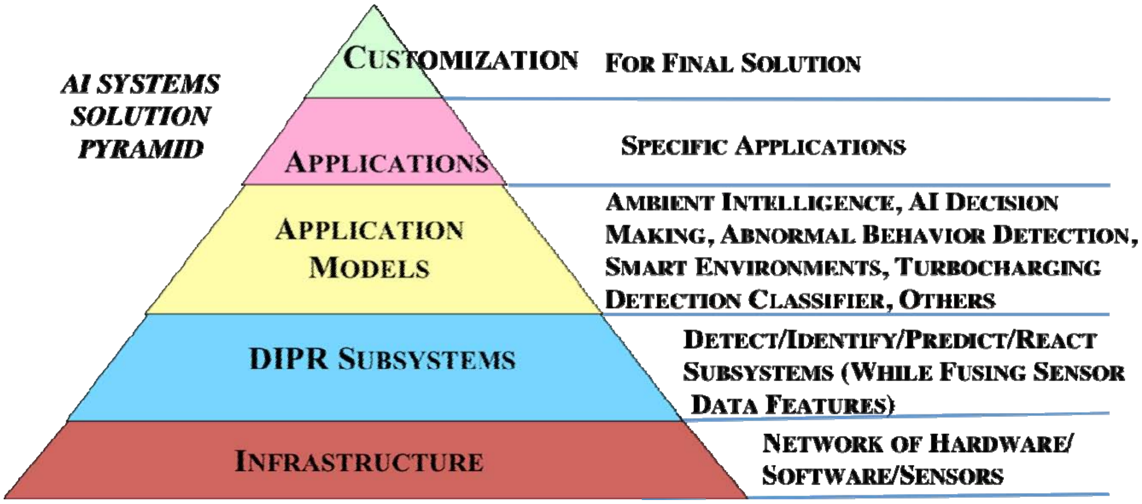

AI SYSTEMS SOLUTION PYRAMID
CUSTOMIZATION
FOR FINAL SOLUTION
APPLICATIONS
SPECIFIC APPLICATIONS
APPLICATION MODELS
AMBIENT INTELLIGENCE, AI DECISION MAKING, ABNORMAL BEHAVIOR DETECTION, SMART ENVIRONMENTS, TURBOCHARGING DETECTION CLASSIFIER, OTHERS
DIPR SUBSYSTEMS
DETECT/IDENTIFY/PREDICT/REACT SUBSYSTEMS (WHILE FUSING SENSOR DATA FEATURES)
INFRASTRUCTURE
NETWORK OF HARDWARE/ SOFTWARE/SENSORS

Figure 7. System of Systems (SoS) Hardware and Software Net-Centric Solution (From Goshorn)

The total solution cannot be realized without the foundational *Infrastructure* (first level of the pyramid) upon which all of the other systems can be built and which allows

[81] David C. Schafer, A Systems Engineering Survey of Artificial Intelligence and Smart Sensor Networks in a Network-Centric Environment, Thesis, Naval Postgraduate School, September 2009.

them to function. This infrastructure comprises the essence of NCO, in that it includes all of the operational elements that must be networked, as well as the means by which they join and remain viable within the network. These elements include the distributed sensors, the communication capabilities, and the platform hardware and software upon which the infrastructure is carried and functions.

The second level of the pyramid is the Detection, Identification, Prediction, and Reaction (DIPR) system, along with its requisite subsystems, that form the heart of the AI systems solution. The DIPR system is a generalized (application independent) software approach that segregates the AI behavior-modeling problem into four discrete intelligence functions to accomplish the AI system's requirements. This system is discussed more thoroughly in the next section, as this particular systems solution level applies most directly to the SE approach for the Watchman POCS.

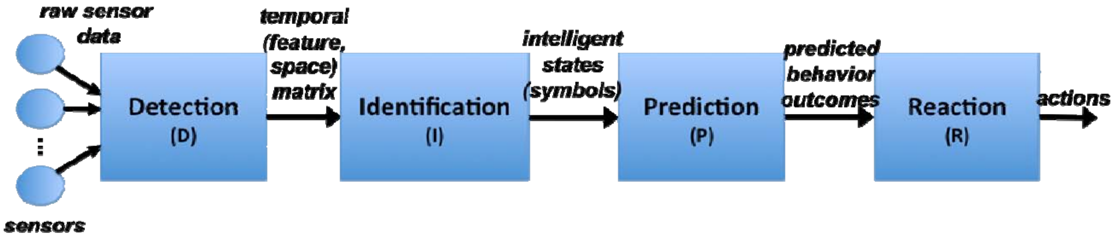

raw sensor data
sensors
Detection (D)
temporal (feature, space) matrix
Identification (I)
intelligent states (symbols)
Prediction (P)
predicted behavior outcomes
Reaction (R)
actions

Figure 8. DIPR Subsystems (From Goshorn)

The Application Models, shown as the third level of the pyramid, engineer DIPR subsystems to perform in a specific rather than generalized solution. Application Models take the general system concepts of DIPR and refine them into more specific solution set ideas that then can form the basis of the actual Applications (fourth level of the pyramid) that are designed to carry out defined model functions. Once the Application Models and the DIPR subsystems are defined within the specific applications, Customization (fifth level of the pyramid) of all level elements for the final solution can be accomplished.[82]

[82] Goshom et al., "Behavior Modeling."

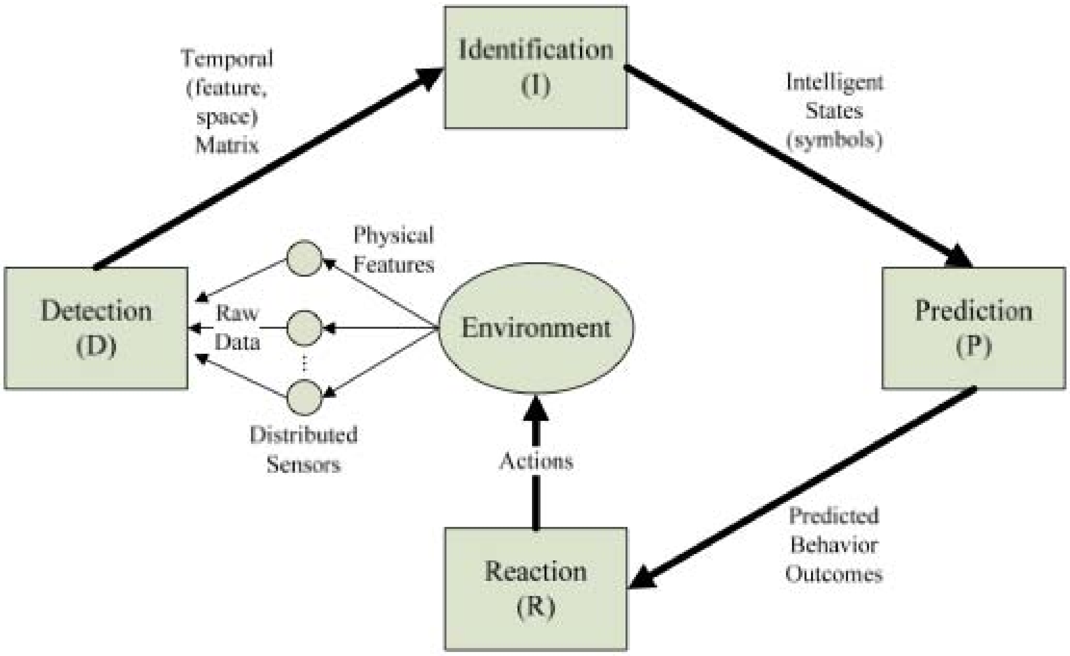


Figure 9. DIPR System Model (From Schafer)

2. The Detect, Identify, Predict, React (DIPR) Model

A concise overview of the generalized DIPR system software approach for AI systems engineering is discussed in this section. Figure 8 depicts the flow of data through the functional processing blocks that represent model subsystems, each explained in the following subsections. Figure 9 presents a visual synopsis of the system, where Schafer relates the DIPR model to a multi-agent system, and the environment shown at the center of the diagram represents that domain in which the DIPR must operate, acquire information from and react to.[83] In the case of the Watchman POCS, this environment is the second floor of Bullard Hall at the Naval Postgraduate School. Operationally, under the DRM outlined in Chapter I, this would translate to the Maritime Domain. As shown, data is obtained from the environment, processed through the Detection, Identification, Prediction, and Reaction subsystems to eventually produce some effect (e.g., predictions leading to reactions, based upon rules of engagement) back onto the environment.[84]

[83] Schafer, A Systems Engineering Survey.

[84] Goshorn et al., “Behavior Modeling.”

a. Detection Subsystem

The function of the detection subsystem is to process raw data input from networked sensors. From this data, the detect function extracts features from the objects sensed. From this input, the subsystem then outputs a temporal feature matrix (Figure 10). A matrix is generated for each time sample. The rows define object feature types. The columns represent disparate spatial regions. Each time a feature is detected in a particular spatial region, the associated matrix element becomes non-zero. That element either becomes 1 to reflect, “detected,” or it becomes a number between 0 and 1 to reflect the confidence level in that feature being detected. In Figure 10, (a) is a simple temporal (feature, space) matrix example where features are binary-valued (black and white). Matrix (b) is a temporal (feature, space) matrix example where features are either real valued and normalized between zero and one (grayscale) or binary (black and white).

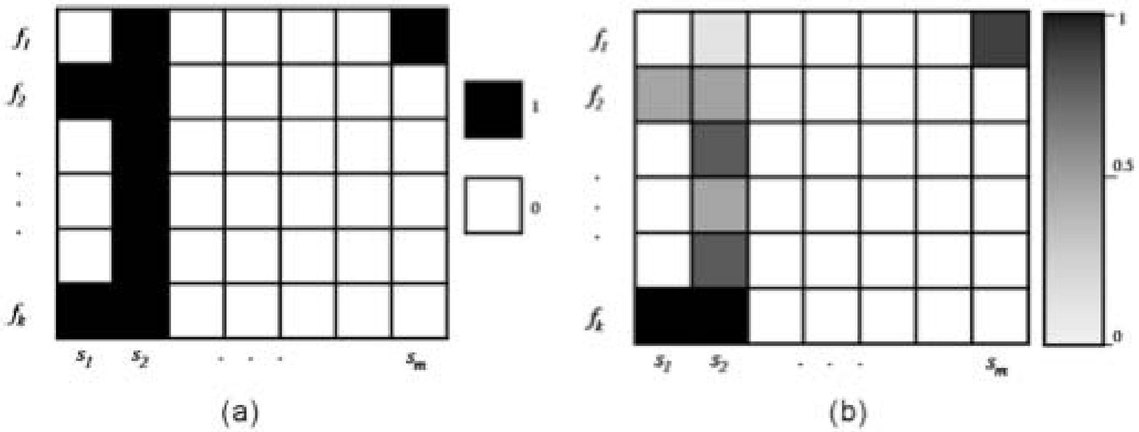


Figure 10. Temporal (Feature, Space) Matrix Examples (From Goshorn)

It is crucial that the detection architecture for the subsystem not only take into account the desired feature(s), ranging from low-level simple features to more complex high-level features, that should be extracted from the data (e.g., visual data), but also the time and space of data extraction for future fusion and correlation of data. For each increment of time, the features that are detected are maintained in all their respective space increments in a 2-dimensional matrix, which then yields a 3-dimensional matrix storing feature values for each time increment, as seen in Figure 11. This sensor data processing of the detect function may be done either locally at the sensor or within a sensor node processing center. This distributed processing is a powerful tool for either reducing bandwidth used and processing power used at the node, in the case of local processing; or utilizing the generally higher processing power of a node center, in the case of node system processing. The systems engineer must be able to understand the capabilities of the system elements in order to properly judge the risks and benefits of each method. Broadly understood, the primary factor in the functioning of the detection sub-system are the“low-level classifiers” that determine the what, when, where in order to categorize and process detected features for extraction.[85]

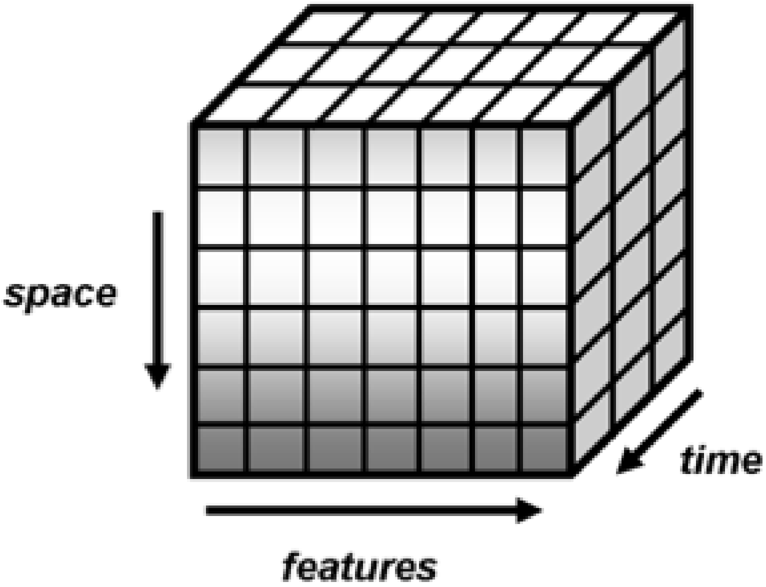


Figure 11. Spatial-Temporal Feature Matrix. (From Goshorn)

b. Identification Subsystem

The primary purpose of the Identification subsystem is to identify specific combinations of fused features in time and space. This fuses the time-space-feature matrix elements, depicted by the 3-dimensional spatial-temporal matrix in Figure 11, which are outputted from the detection subsystem. When the fusion function identifies a

[85] Goshorn et al., “Behavior Modeling.”

particular state of fused spatial-temporal features in one time interval, an "intelligent state" or "symbol" is generated. The fusion functions are called "intelligent rules," which are usually defined by the user, depending upon the application. An example of a simple intelligent rule could be a Boolean expression in which ranges of features are fused according to a logical "and" descriptor (i.e., f1(t_1,s_1) and f2($t_{2,}s_2$) and f3(t_3s_3)). The clauses in the expression represent the ranges of time/space values for each feature, for which the defined rules then describe a particular intelligent state (symbol). For any given time interval, the rule is satisfied, and the corresponding intelligent state (symbol) is identified when feature 1, f_1, is detected at time t_1, in spatial region s_1 *and* feature 2, f_2, is detected at time t_2, in spatial region s_2, *and* feature 3, f_3, is detected at time t_3, in spatial region $s_{3.}$ The output from the "Identification" subsystem is the identified state symbol.

Therefore, the intelligent states (symbols) at each time step are generated simply when the desired features take on the desired range of values for specific spatial regions, carried out through the Fusion (Intelligence Rules) Module. The Fusion Module is shown in the second stage of the "Identification" subsystem diagram, Figure 12. The creation of intelligent states (symbols) in rule definition is governed by the relevance of those symbols in ultimately capturing an aspect of a defined behavior. Thus, the Fusion Module identifies intelligent states (symbols) by recognizing only the desired fusion (intelligent rules) of temporal feature values at only desired spaces.[86]

[86] Goshorn et al., "Behavior Modeling."

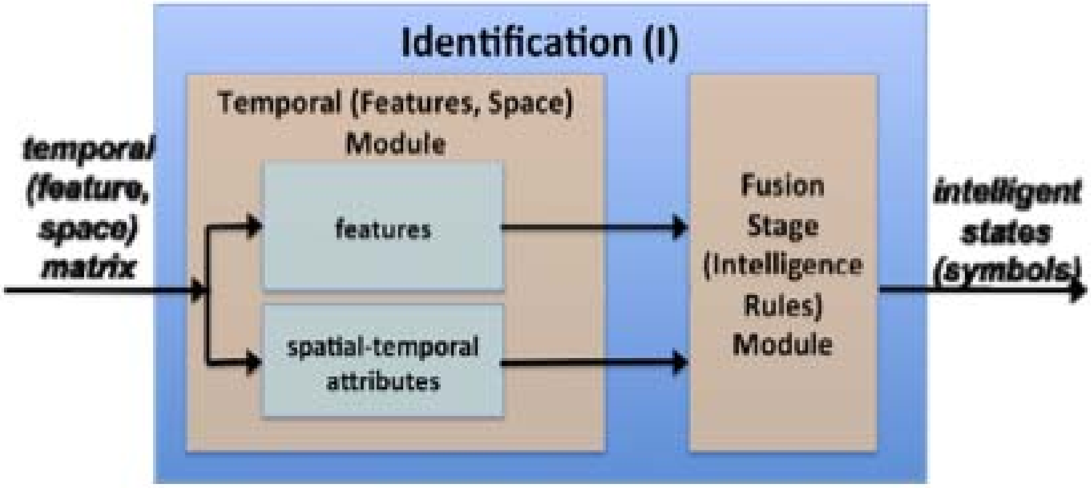

Identification (I)
Temporal (Features, Space) Module
temporal (feature, space) matrix
features
spatial-temporal attributes
Fusion Stage (Intelligence Rules) Module
intelligent states (symbols)

Figure 12. “Identification” subsystem of the DIPR system. (From Goshorn)

As has been explained, for every time-feature-space matrix inputted, there is an intelligent state (symbol) outputted when the fusion (intelligent rules) are met. If the rules are not met, a null state (symbol) is outputted. The use of the null state (symbol) is key to the “Prediction” subsystem behavior analysis function. For example, the system loses track of a contact for a certain time, or has not initiated a track sequence (first module of “Prediction”) on a contact.

In addition to identifying the intelligent state (symbol), the “Identification” subsystem creates a three dimensional matrix (feature, space, time) which stores the 2D (feature, space) matrix by storing the feature values at different spaces in the matrix at every time increment, as seen in Figure 9. This three dimensional matrix is useful for intelligent states (symbols) extracted from incremental features in space over time. For example, assume feature (f_k), estimates velocity derived from the difference in space and time of the contact being tracked.

The creation and storage of this three dimensional matrix is also very important for the learning aspect of DIPR. Automated learning is a vital function in order for the system to describe previously unseen intelligent states (symbols) and behaviors.[87]

[87] Goshorn et al., “Behavior Modeling.”

Lastly, the storage of the values in the three-dimensional matrix form is critical for memory and computational efficiency. Other fusion and identification systems store feature data in a list or database formats, which are extremely demanding in memory and processor capacity. Matrix data storage and manipulation is essential to allowing DIPR functionality to be integrated on smaller, disadvantaged platforms, as well as to enhance DIPR distributed processing.

c. Prediction Subsystem

The "Prediction" subsystem is the heart of high-level AI, behavior modeling, and analysis and is understandably the most complex subsystem of DIPR. The methodology behind this subsystem is what enables the effective systems integration of automated behavior analysis (ABA). The subsystem receives the identified intelligent states (symbols) as input from the "Identification" subsystem and outputs predicted behavior outcomes. This stage executes high-level behavior classification, as well as prediction of the corresponding contact behavior. In order to accomplish these functions, sequences of intelligent states (symbols) are first classified into behaviors and then these behaviors are inferred (predicted) to be one of the "known," or pre-defined prediction outcomes or potentially the "unknown" prediction outcome. The "Prediction" subsystem is comprised of two modules, shown in Figure 13.[88] Via the DIPR model, the prediction function is executed as fused features (symbols) are tracked over space and time, as opposed to tracking the features themselves, as in standard AI approaches. The latter, more typical feature-tracking approach, which is processor and bandwidth intensive, presents the greatest challenge, and indeed restriction, to high-level AI behavior analysis. The following explanation of the functions of the Prediction modules will demonstrate how high-level AI behavior analysis becomes possible with DIPR, in an MDA, or any mission environment.

[88] Goshorn et al., "Behavior Modeling."

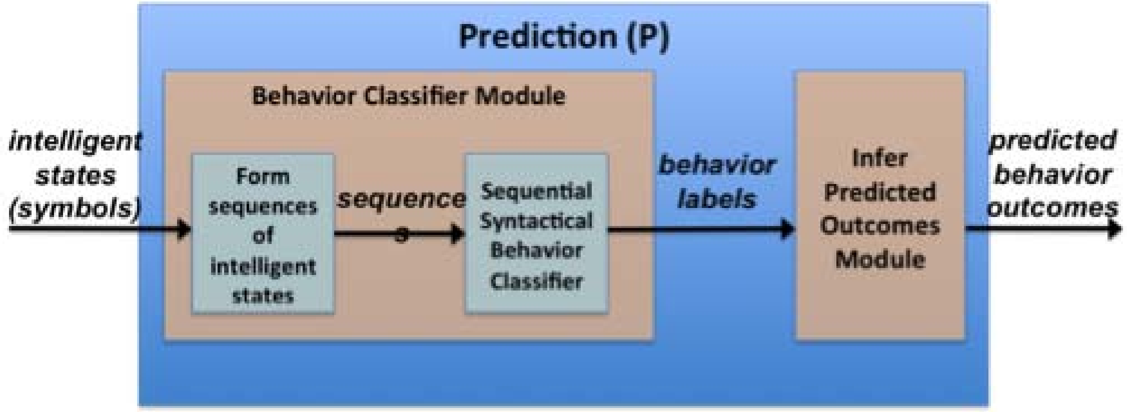

Prediction (P)
Behavior Classifier Module
intelligent states (symbols)
Form sequences of intelligent states
sequences
Sequential Syntactical Behavior Classifier
behavior labels
Infer Predicted Outcomes Module
predicted behavior outcomes

Figure 13. “Prediction” subsystem of the DIPR system. (From Goshorn)

The initial function within the Behavior Classifier Module (seen in Figure 11) forms sequences of intelligent states (symbols) in some type of intelligent state (symbol) buffer. The second stage of the module is the sequential syntactical behavior classifier, explained in subsection (1). The Behavior Classifier Module outputs a behavior label. This label is either one of a number of pre-defined or “known” behaviors or an unknown behavior. The behavior will be labeled as “unknown” if the sequence was not classified as any of the possible “known” behaviors. The behavior labels are then input to the second stage, the Infer Predicted Outcomes Module. Just as the name suggests, this stage infers one of the predicted behavior outcomes. The inferred predicted outcomes are based on circumstances specified by the operator, environment, and application.[89]

(1) Behavior Classification Module. The Behavior Classification Module (BCM) enables ABA systems integration through the efficient processing of sequenced symbols, derived from features, rather than the processing of features themselves, which, to reiterate, is very memory and processor intensive. This powerful processing tool is found in the BCM second stage, the Sequential Syntactical Behavior Classifier (SSBC), which outputs a behavior label upon processing of the input sequence of symbols. If a behavior is not classified as one of the predetermined behaviors, it will be given the unknown behavior label. The behavior labels are

[89] Goshorn et al., “Behavior Modeling.”

predefined with user input to the system developer, however this predefinition function has the ability to be automated, which is highly desirable for operationally effective usage. Ideally, automated behavior definition would occur in two primary ways, through automated behavior learning, or intelligence data updates (data "push"). To understand the approach to defining behaviors, a brief discussion is provided, excerpted from Goshorn, et al.

Behaviors are represented with a syntactical grammar-based approach, where each predefined behavior is modeled as a statistical and cost-weighted Finite State Machine (FSM).

Let the set of all possible intelligent states (herein after referred to as "symbols"), which again, are derived from detected features and created in the "Identification" subsystem, compose an alphabet Σ. An example alphabet is Σ = {a_1, a_2, . . . a_N } where each letter represents an identified symbol. This is a simplified example, as an "alphabet" of symbols could be comprised of members that contain any combination of letters and numerals, which could represent certain feature attributes. Defining more complex and meaningful symbols was a large portion of the Watchman ABA research effort, and is discussed in Chapter V. A behavior can then by defined through a set of syntax rules that combine the elements of Σ into a particular sequence. For example, if *a* is a symbol and we want to model a certain behavior, denoted B_k, which describes when only the intelligent state, *a*, is observed over a period of time, we would model it with the following FSM language, shown in Equation [1], where symbol *b* signifies the intelligent state of terminating the sequence.

$$B_k = \begin{bmatrix} S \rightarrow aQ_1 \\ Q_1 \rightarrow aQ_1 \\ Q_1 \rightarrow bF \end{bmatrix} \qquad [1]$$

In other words, the FSM reads the observed sequence, and the sequence is then classified according to the behavior whose corresponding FSM accepts the sequence. It is certainly likely that a sequence could not be accepted by any of the predefined behaviors, due to the fact that systems are not completely predictable and reliable. This unreliability could be attributed to errors in feature detection or collection, low-level feature classification, or a behavior pattern statistically deviating from what has been defined.

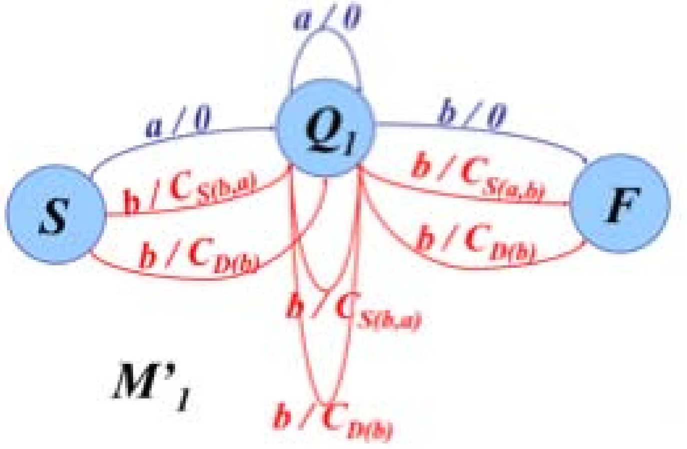


Figure 14. Augmented FSM M’1 of behavior Bk. The augmented syntax rules are in red, and original syntax rules in blue with zero cost (From Goshorn)

To cope with these inevitable deviations, the FSMs for each defined behavior are augmented with “error production” rules. For example, suppose symbol *a* and *b* are two separate symbols and due to the feature detection errors mentions previously, the symbols are sometimes confused for one another and are interchanged. Let S(*a*, *b*) refer to substituting the true symbol *b* for the mislabeled symbol *a*, assigning an associated cost $C_{S(b,a)}$ for doing so; and D(*a*) refers to deleting a mislabeled symbol *a* with an associated cost $C_{D(a)}$. The corresponding modified FSM, M_k is shown in Figure 14. The costs for these substitutions and deletions must then aggregated to determine a “distance” from the observed sequence to the predefined behavior sequence.[90]

Shown by Equation [2], Goshorn calculates this “distance” in the following manner:

[90] Goshorn et al., “Behavior Modeling.”

Let the set of possible symbols be $\Sigma = r_1, r_2, \cdots, r_N$, where N is the total number of predefined intelligent states (symbols). With this, the distance from an observed sequence of symbols *s*and a predefined behavior B_l is given by:

$$d(\mathbf{s}, B_l) = \sum_{i=1}^{|\Sigma|} \sum_{j=1}^{|\Sigma|} C_{S(r_i, r_j))} n_{S(r_i, r_j)} + \sum_{i=1}^{|\Sigma|} C_{D(r_i)} n_{D(r_i)} \quad [2]$$

where $n_{S(r_i,r_j)}$ is the number of substitutions of the true symbol r_j for mislabeled symbol r_i, and $n_{D(r_i)}$ number of deletions of the mislabeled symbol r_i. Assuming K behaviors, each behavior and its associated FSM are augmented a priori so that any sequence of intelligent symbols is accepted by each behavior, but with a total cost. The augmented behaviors are denoted by $B_1, B_2, ... B_K$, and their ‴K corresponding augmented finite state machines are denoted by $M'_1, M'_2, ... M'_K$. ′An unknown sequence *s* of intelligent state symbols is then parsed by each M_l , ′with a cost $d(s, B_l)$. The sequence s is then classified as the behavior B_g, where $B_g = \min_d(s, B_{\prime l}, l = 1, 2, \cdots, K)$, and is the behavior to which it is most similar. Therefore, sequences of symbols are classified based upon Maximum Similarity Classification (MSC) as seen in Figure 15.[91]

[91] Goshorn et al., "Behavior Modeling."

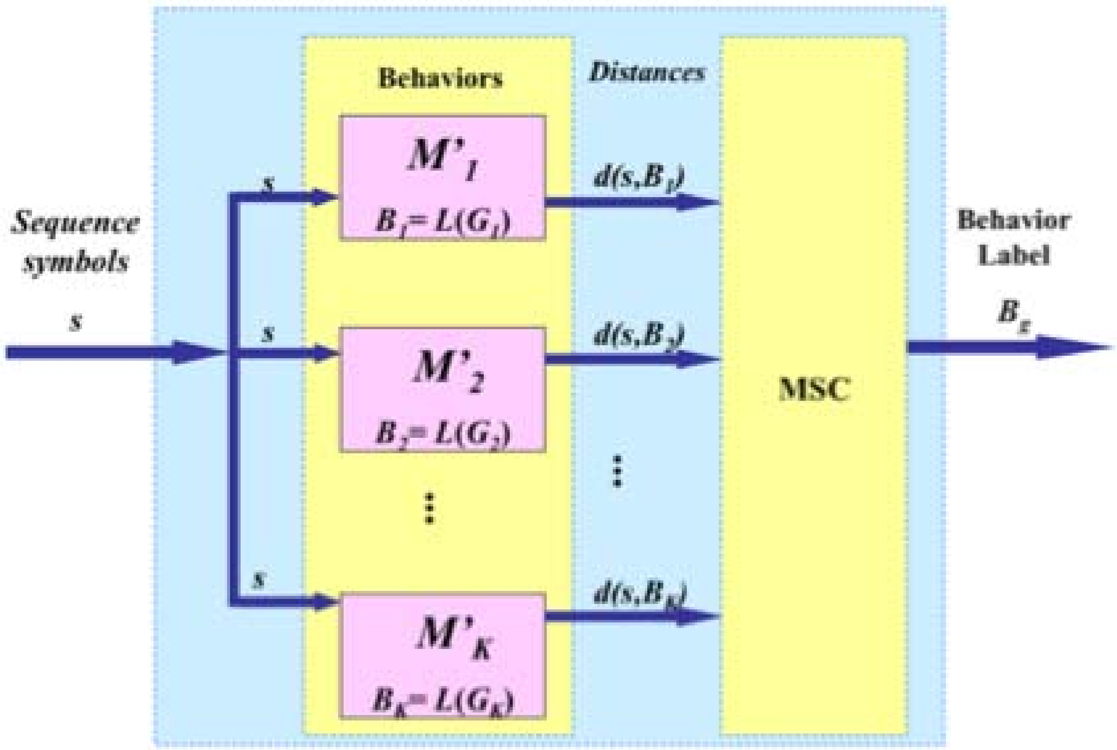


Figure 15. Maximum Similarity Classification (MSC) (From Goshorn)

Automating behavior syntax rule costs is a significant part of the behavior modeling process and is discussed generically in this section. Cost automation for the increased complexity of the multi-sensor, multi-node, distributed processing environment of the Watchman system was a significant challenge; however, it was met with a rather elegant solution. The details of this work are discussed in Chapter V.

In the general case, the costs can be automated by first considering the inherent possibility of confusing two symbols. This possibility of confusion between two symbols is valuated with a certain probability between 0 and 1. For example, assume that only three predefined symbols, *a*, *b*, and *c* can be identified, and that six possible confusions may occur. Let P(*a*|*b*) equal a probability that a true symbol *a* might have been confused with the symbol *b*. The probabilities of all possible symbol pair confusions can be represented in a Symbol Confusion Probability matrix, shown in Table 5.

	Labeled: *a*	Labeled: *b*	Labeled: *c*
True: *a*	*P(a|a)*	*P(a|b)*	*P(a|c)*
True: *b*	*P(b|a)*	*P(b|b)*	*P(b|c)*
True: *c*	*P(c|a)*	*P(c|b)*	*P(c|c)*

Table 5. Symbol Confusion Probability Matrix (From Goshorn)

In an ideal world, the Symbol Confusion Probability Matrix would be the Identity matrix, where the diagonal entries are one and the off-diagonal entries are zero. If there is no prior knowledge about the inherent confusion probabilities between symbols, then the identity matrix can be used as a default valuation mechanism for automating the costs. Otherwise, prior knowledge of symbol confusion probabilities that can be derived or discovered should be utilized to more accurately define the Symbol Confusion Matrix for the particular application and environment.

The substitution costs are defined as the log of the inversion of the conditional probability entries in the Symbol Confusion Matrix.[92] This is shown in the following example, depicted in Equation [3]: The cost for substituting the true symbol *b* for the observed symbol *a* is a function of the inversion of the probability that the true symbol *a* is confused with the observed symbol *b*. By representing the conditional probabilities P(truesymbol = *a*|observedsymbol = *b*) as P(*a*|*b*), then the cost for substituting a misidentified symbol *a* with the true symbol *b* is:

$$C_{S(a,b)} = 10log_{10}(\frac{1}{P(a|b)}) \qquad [3]$$

Intuitively, if it is highly likely that the true symbol *a* will be confused with symbol *b*, where P(*a*|*b*) is high, then the cost for substituting symbol *a* for

[92] Goshorn et al., “Behavior Modeling.”

b, CS(*b,a*), is clearly low. Also, if the probability P(*a*|*b*) is nonzero (which is likely, given an imperfect world), then P(*a*|*a*) is less than one, since, from Equation [4]:[93]

$$P(a|a) = 1 - P(a|b) - P(a|c). \qquad [4]$$

(2) Infer Predicted Outcomes Module. The "Prediction" subsystem first stage, BCM, outputs behavior labels, which are then input for the second stage, the Infer Predicted Outcomes Module (IPOM). The IPOM fuses behavior labels with particular predefined application or environment knowledge, according to predefined inference rules. For example, if a behavior label output was "abnormal behavior by group of small surface vessels, excessive speed and common heading towards High-value Unit," an inferred predicted outcome could be: "Swarm Attack." For this example, the inference rule would be: if the behavior label "abnormal behavior by group of small surface vessels, excessive speed and common heading towards High-value Unit" is given, the inferred predicted outcome is "Swarm Attack." The inferred outcome, "Swarm Attack," would compel a specified action, which is discussed in the next DIPR subsystem of "Reaction." Certainly, any inference rules applied would be domain and application dependent and would most likely require some type of automated learning and updating for operational implementation.

d. Reaction Subsystem

The primary purpose of the "Reaction" subsystem is to create autonomous (though not necessarily without human intervention or oversight) actions in response to the predicted behavior outcomes. In other words, the goal of the Reaction subsystem is to automate those governing reactionary systems, such as the rules of engagement (ROE), to reduce reaction times, and hopefully eliminate, or at least greatly reduce, the human error inherent in those systems. The inputs to "Reaction" are predicted behavior outcomes that stem form the "Prediction" subsystem, with outputs of actions. These actions are application and domain dependent; therefore, reaction rules are defined as a function of

[93] Goshorn et al., "Behavior Modeling."

the inputted inferred predicted outcomes, which are also dependent on the same unique domain and application factors. The actions and the rules that define them must have the capability of being adapted and learned (either system-wide or user-wide) for their specific operational environments to ensure appropriate actions take place. In the example given above of an inferred predicted outcome of "Swarm Attack," the pre-determined action could be "Set General Quarters and launch counter attack." This type of response would be determined by such discrete environmental factors such as Rules of Engagement (ROE), weapons posture, maneuvering constraints, Operational Orders (OPORDS), etc. Not only are these factors environment dependent, but also often situation dependent, and can change over time, which requires constant updating for optimal implementation, which increases the need for automation in this arena, as well.[94]

The Watchman system project also integrated an automated response capability to demonstrate the "Reaction" concept of DIPR. This capability was in the form of a Wireless Smart Sensor Network (WSSN) of autonomous robot "agents." A full description of the systems integration and demonstration of this capability is elucidated in Chapter V.

B. SYSTEMS ENGINEERING PROCESS MODEL

The WMSE POCS Systems Engineering and Integration project followed a somewhat modified version of the classical "Vee" model of Systems Engineering depicted in Figure 16. This project included all aspects of design, build, and test of a fully functioning system, to include both hardware and software components, throughout several phases. The project team utilized a modular, object-oriented, agile approach in the software development of the system, seeking to pull the best practices from various software development processes throughout the course of the project. The architecture, design, build and test of each sub-system functional area were iterated numerous times, both within and between project phases. These iterations included decomposing the requirements, functions, and processes, to integrate each subsystem unit into each

[94] Goshorn et al., "Behavior Modeling."

subsystem, and then integrating the subsystems into the complete system. This process was re-iterated for each subsequent upgrade and improvement to the various system segments. The segment-lead for each segment/subsystem oversaw and managed the planning and execution of each functional area, guiding their team through the steps of the SE process.

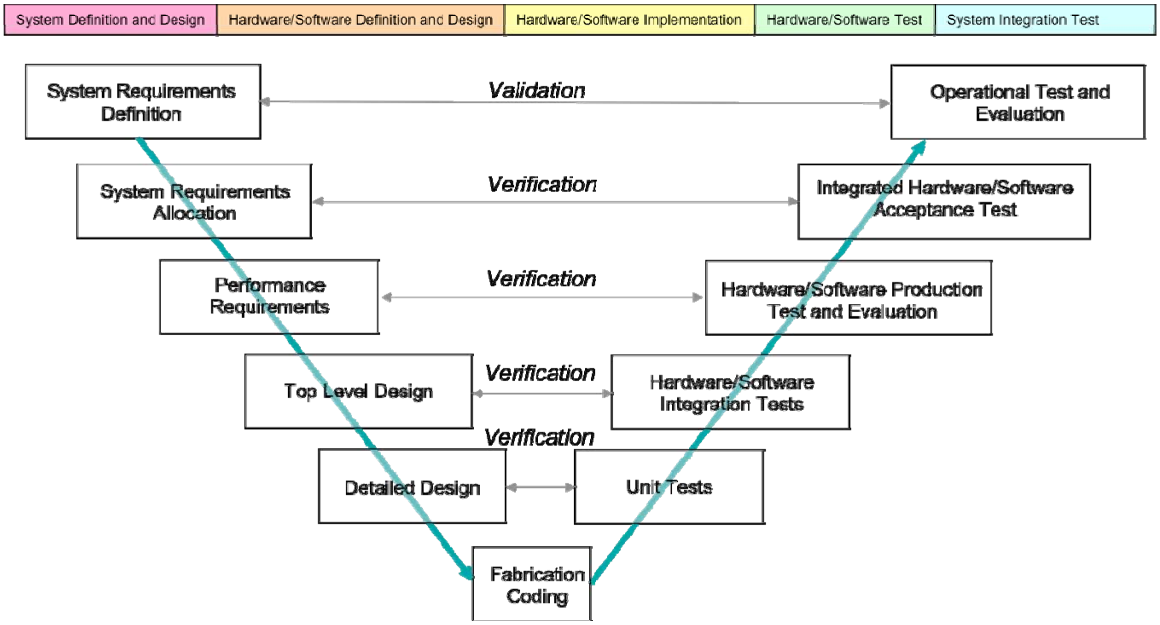

System Definition and Design
Hardware/Software Definition and Design
Hardware/Software Implementation
Hardware/Software Test
System Integration Test
System Requirements Definition
Validation
Operational Test and Evaluation
System Requirements Allocation
Verification
Integrated Hardware/Software Acceptance Test
Performance Requirements
Verification
Hardware/Software Production Test and Evaluation
Top Level Design
Verification
Hardware/Software Integration Tests
Verification
Detailed Design
Unit Tests
Fabrication Coding

Figure 16. Systems Engineering "Vee" Process Model (After Whitcomb)

C. STEPS OF THE "VEE" FOR THE WMSE POCS

The following is a description of the activities, events, and milestones that were met during each stage of development depicted in the SE "Vee" model for each project design iteration, or "phase". Some stages of the process, as is shown and described, did not match identically to the model, as the model was followed more as a guideline than a mandated process. Those stages that were appropriate to the scope and nature of this project were adopted, though some with modification, and others were eliminated, such as "Operational Test and Evaluation," given that the system was developed as a proof of concept and not for an operational environment. Stages often ran concurrently, and were iterated, as necessary in each project phase, until system units and subsystems were completed and tested.

As mentioned, the project consisted of several phases, that each progressed through the steps of the "Vee" model. Throughout all of these phases, though a SE process was followed, it was not always in precisely the same manner, as each phase was a unique design experience. The phases were: Initial Development, Engineering Change Order (ECO) Implementation, Wireless Smart Sensor Network (WSSN) Integration, and ECO refinement. In the process description that follows, those aspects of each phase that required a deviation or alteration to the "standard" or original process is discussed, to illustrate the flexible nature of the SE process model and how that flexibility was exploited for successful project execution.

1. Requirements Analysis and Definition

As the system was first being conceptualized, the requirements that were first analyzed were derived from group discussion and consensus among the project team as to what a generic surveillance system would need to accomplish in order to be effective. Many varied real-world operational needs were discussed, as well. This process was somewhat "having the cart before the horse," however; new technologies and technologically innovative methods must often find a "home" within the universe of operational needs before they are found to be germane and useful.

As the project progressed and individual students began to branch out into differing areas of study related to the Watchman capability, other real-world requirements began to be looked at for relevance to the project. For this thesis, the MDA JIC capability gaps addressed in the DRM and OV-1 discussion in Chapter I were analyzed and defined in terms of the possibility of integrating a Watchman-like behavior analysis capability to fulfill those gaps. The analysis of these requirements was conducted in much the same process as the generic laboratory requirements, consisting of functional analysis and allocation and hierarchical functional decompositions. This analysis, however, was more in depth, and included identification and decomposition of operational activities, processes, operational nodes, resources, components, interfaces, links, and more that were all linked within a system architecture definition that is explained in Chapter IV and thoroughly detailed in the Watchman MSE System Description Document (SDD), in the Appendix.

2. Architecture (Top Level) Design

This concept development stage soon evolved into a requirements definition stage, where the required functions were identified and decomposed into a functional hierarchy that formed the basis of the Watchman subsystem groups that is explained in detail in Chapter V. The project team was then divided amongst the groups, each with a team lead, and then each team conducted further requirements definition and analysis through continued functional analysis and allocation, which then led to individual group functional hierarchies. Those hierarchies then formed the basis of the functional architectures, which then became the foundation of the detailed system specifications from which the initial system designs were derived. These specifications are also outlined in Chapter V. WMSE POCS architecture definition is also discussed in detail in Chapter IV.

The POCS architecture was refined, expanded, and improved for each subsequent phase of the project. After the architecture, development and integration were successfully demonstrated in the POCS, the flexible nature of the architecture allowed for the incorporation of ECOs, and WSSN functions. Architecture refinements included a more thorough mapping to requirements, along with revised functional, operational and process decompositions, which revealed redundancies, inconsistencies, and inefficiencies in requirements and the designs that were addressed in the ensuing project phases.

3. System (Detail) Design

With all project phases, the subsystem designs all went through multiple iterations, being drafted and then analyzed against the requirements, until prototype units could be developed. These prototype designs contained enough detail to allow the units to perform the base functionality delineated in the architecture, but with enough flexibility to allow for changes, should the initial design not fully meet the functional requirements. Additionally, each team had to coordinate with the other teams ensure that their designs included the appropriate hardware and software interfaces and data structures so that the subsystems could be tested individually, as well as ultimately be integrated and function as an overall system.

During this stage, requirements were further refined, as feasibility of certain requirements, given the hardware, software, and time constraints began to materialize. Those requirements that were deemed unattainable in the first design iteration were either eliminated as redundant or unnecessary to meet the initial conceptual need, or were deferred so that they might be accomplished in a future system upgrade. For those requirements that were deferred, allowance was made in the design, in terms of an Open System Architecture (OSA) to facilitate the future design of the required functionality.

4. Fabrication / Coding

Source code for each individual element had to be developed from the ground up, or modified and/or adapted from existing COTS, open source, professor-provided code. Implementing each aspect of D-I-P-R meant coding effort for the various subsystem elements, whether for the subsystem application itself or for the interfaces between subsystems. Additionally, special code had to be written that allowed COTS hardware and software to function within this particular project application architecture. For example, for the *Detect* subsystem, the WiLife cameras produced processed video files in the Windows Media Video (.wmv) format as a function of their embedded software. However, the software team had to write code that could use those processed files downstream in the system to automatically extract contact features that would be inputted into the next processing stage to meet *Identify* subsystem functional requirements. Additionally, the proprietary video processing did not allow for capture and further processing of live, streaming video data. This constraint was only discovered during the coding of the contact-tracking program (described in Chapter V), and led to a design change “work-around” code-test-fix-test iteration toward the end of the first phase. It is this type of flexibility exercised within the model on numerous other occasions that testify to the efficiency of the process.

Hardware installations and modifications were no less challenging. Camera placement, computer installation and set-up, network wiring and connection to all the processing nodes, as well as all nodes to each other was a detailed, painstaking task. Nodes were individually set up, tested, and then interconnected one by one, to ensure video collection and data transfer quality within nodes and across the network.

Hardware and Software for the ECO and WSSN integration phases were very similar in process, even as the implementation was quite different. The ECO phase required extensive coding of software upgrades, additions, and "fixes" to the initial Watchman functionality, while the hardware changes were the addition of an entirely new node construct with new types of sensors. The WSSN phase was unique as it introduced a new type of data transfer (Bluetooth) to the system, along with yet another new type of sensor node, wireless smart "agents," in the form of autonomous, self-navigating and self-prioritizing robots. The details of the changes for both of these phases are discussed in Chapter V. Despite the radically different achievements for each phase, the consistency of the process ensured reliable results.

5. Unit Testing

Testing for hardware and software was conducted incrementally as the system designs were implemented. Each element was tested individually, then as part of each subsystem. As each subsystem was tested successfully, subsystems were incrementally joined, on to another, and tested as integrated components, verifying the designs and network infrastructure. Much of the testing was conducted in parallel with iterative design-change coding (code-test-fix-test), in order to provide a more agile development approach. There was much trial and error in this method; however, it allowed for very rapid prototyping in each phase that eventually led to the ultimate initial design for each subsystem that could then be integrated into the full system.

6. Integration Testing

As mentioned previously, units were first individually tested, using dummy input and output that mimicked adjoining units. Once these tests were successful, subsystems were then joined from units and subsystem testing was completed, using the same dummy input and output method, mimicking adjoining subsystems. At the successful completion of these tests, subsystems were joined, one at a time, testing all interfaces and data transfers, as well as individual process behaviors utilizing real data. Code that interfaced between separate systems was first tested on a small scale; corrected or adjusted as needed, and then more comprehensive, end-to-end testing was conducted, measuring performance and losses against the derived specs.

7. System Testing

Eventually, all systems were joined and tested as a full system, again using a test-fix-test methodology. System demonstration (end-to-end, dry-run and for score) tests were conducted on all interfaces and individual interface issues (e.g., database writing not occurring as expected/needed, files not written to the correct locations, database queries incomplete or not running properly, etc.) were corrected.

8. Acceptance Testing

The acceptance tests for each subsystem, as well as for the entire system, demonstrating full system functionality were conducted in order to validate the system’s performance against the requirements was conducted at the end of the initial system development process, as well as each subsequent development phase process. The procedures for all of these acceptance tests are described in detail in Chapter V. These system demonstrations for all phases were conducted on several occasions, including several “open house” demonstrations, and for numerous individuals, from NPS faculty, staff, and students, to curriculum and project sponsors, systems command representatives, science and technology community members, as well as members of the technological media. All commented on the innovation of technology represented by the project, the uniqueness of the design and implementation approach, and the wide array of mission needs, including MDA, that the system concept had the potential to fulfill.

This chapter discussed the SE approach utilized and how that SE process led to the development of the Watchman Maritime Smart Environment (WMSE) proof of concept system (POCS). The chapter began with a generalized view of an SE approach to

Artificial Intelligence, in particular Behavior Analysis, and concluded by describing the steps of the SE process, as they pertained to each phase of development of the POCS.

Chapter IV discusses the various architectures that were created as part of the SE process. The chapter describes the architecture development process as a subset of the overall SE process for system alternative analysis and design development, as well as shows how the resultant architecture was ultimately applied in developing the WMSE POCS.

IV. MSE SYSTEM ARCHITECTURE

This chapter is intended to describe and present the functional, process and operational architectures that were produced as part of the functional analysis and allocation steps of the SE process. The chapter sections describe the initial and integration architecture development processes as subsets of the overall SE process for system alternative analysis and design development. The chapter further shows how the resultant architectures were ultimately applied in the design, building, coding, integration, and testing of the autonomous WMSE POCS that is representative of an MDA "smart" environment.

A. INITIAL ARCHITECTURE DEVELOPMENT

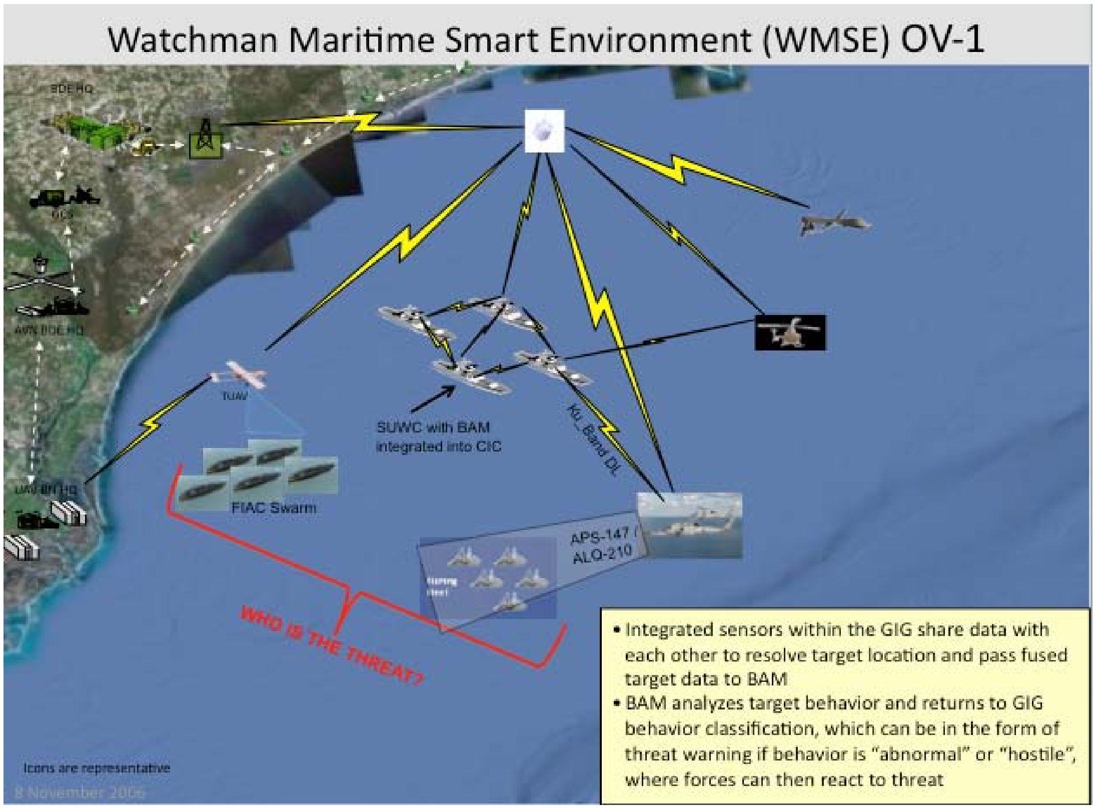
This section discusses the development of the initial architecture for the WMSE POCS, which was created in more of a generalized operational context to allow for proper requirements analysis and functional allocation. The reader should recall from the previous chapter that these steps are on the left upper, or beginning, side of the SE "Vee" model, and are essential building blocks for any SE design and development project. The section includes definitions of the Operational, Functional, Requirements, and Physical architectures, and then concludes the initial architecture development.

1. Operational Architecture Definition

The first step in creating the architecture for the Maritime Smart Environment (MSE) was to define the top operational activity, and the missions that activity was expected to perform, within the scope of this particular part of the entire system. In order to accomplish this, it was necessary to scope and bound the area of responsibility and mission focus for this part. It was therefore determined that this part of the architecture, for this thesis, would deal only with the activities required to perform behavior analysis on selected target data which would be provided to the module in some type of processed, fused form. The module would then perform the analysis and then output a classification and, if required, some type of alert or warning to the user(s) to provide actionable intelligence about the target(s) that had been input. The actual sensors, sensor platforms, their integration and method of data fusion are considered external to the module and beyond the scope of the architecture. Likewise, whatever decisions, plans, actions, movements, or engagements occur as a result of the output of the module are also considered out of scope and will not be addressed in the architecture.

With the activity thus bounded, the main activity was named Maritime Behavior Analysis, and the mission that this activity achieves is Maritime Domain Awareness. This Mission was then decomposed into four constituent missions, derived from two pillars of the Naval Power 21 document referenced in the DRM, which are Sea Shield, and FORCEnet. The edits to remove the missions that were deemed not applicable could not be done in the table depicted in the DRM, given the format of the Naval Power 21 artifact used in the document. However, the sub-capabilities used to define the missions achieved by the overarching activity are: Provide Self-Defense Against Surface Threats, Protect Against Terrorist Threats, Detect and ID Targets, and Provide Cueing and Targeting Information.

The CONOPS for these missions, being accomplished by the Behavior Analysis activity must, by necessity, be somewhat broader than just the BAM itself. The BAM, which is an instantiation of the *Predict* subsystem from the DIPR software application model, must be understood to operate in a larger context of a force protection schema, which consists of sensors, high-value units, communication, command and control systems, and a network in which all of these members communicate and operate. This schema is depicted in the OV-1 diagram in Figure 17:


Watchman Maritime Smart Environment (WMSE) OV-1

- Integrated sensors within the GIG share data with each other to resolve target location and pass fused target data to BAM
- BAM analyzes target behavior and returns to GIG behavior classification, which can be in the form of threat warning if behavior is "abnormal" or "hostile", where forces can then react to threat

Figure 17. WMSE OV-1

As defined in the DRM, the concept is that there is some type of high-value maritime asset, such as an oil platform or Surface Action Group (SAG). This asset must be protected, and is therefore nested within a network of internal and external, organic and inorganic, manned and unmanned, sea-based and land-based sensors, which provide a Common Operational Picture (COP) of the sea surface around the high-value unit(s). This sensor network has the ability to fuse all of the sensors' data flooding in to detect, track, and, to a certain extent, classify the surface vessels in the vicinity of this unit(s). Humans in the loop of this sensor network monitor these tracks, however, their dense number and irregular, indiscernible, and unpredictable movements prevent an early and/or accurate detection of threat-like activity by the many contacts, which must be monitored in the COP. It is at this point that the BAM becomes relevant. As the data is collected, fused, and disseminated, it is passed within the network to a particular friendly unit, which has the BAM processing capability on board. All of the fused track data is processed through the BAM and compared to its many friendly, neutral, and threat behavior models, where it is statistically analyzed, looking for anomalous behavior. Among the many fishing vessels going about their normal routines and other various surface vessel traffic, the sensors pick up a group of vessels moving in some type of formation, apparently approaching the SAG. Assuming this is a threat and taking lethal action could cause an international embarrassment if incorrect. However, assuming this behavior is no threat could delay critical seconds in responding, which could lead to a devastating loss of ship and crew. It is just this type of scenario that the BAM is meant to unfold, providing decision-makers with the crucial pieces of timely information to discern truly hostile from innocent behavior, and then to be able to take appropriate action.

The Operational Node that performs the overarching operational activity is the Surface Warfare Commander (SUWC). It is this operational role that is best suited to execute the activity of maritime behavior analysis since this entity will be the primary consumer of the information that the BAM would provide and would also be the principal decision-maker acting upon that information. There are many other external organizations, which would both feed into, as well as extract from the BAM. Any platform or entity that collects, collates, fuses, and disseminates sensor data would be an organization that could have an interface with the BAM. Additionally, any organization that would have the responsibility to act upon potential threats to maritime domain assets would be a consumer and would therefore require some type of collection or dissemination interface with the BAM. These types of interactions have been depicted in the architecture as information inputs and outputs. Due to the sheer volume of potential donors and customers interfacing with the BAM, only those information artifacts were discussed. External system sponsors will have the ability to see these inputs and outputs and then make recommendations as to how their systems' architectures might tie in with the BAM's

We will now discuss the actual operational architecture structure, with its primary operational activities, tasks, and the metrics that have been chosen to eventually evaluate those tasks in a modeling scenario. The initial architecture of the system was completed using ViTech's CORE software by entering and mapping the system's architecture elements, including the operational, functional, and requirements decomposition, as well as all the required linkages, into the program for analysis and documentation. The functional allocation and analysis began by looking at the required operational activities that would be needed to complete the DRM, according to the OPSIT parameters. Figure 18 shows the first level of decomposition.

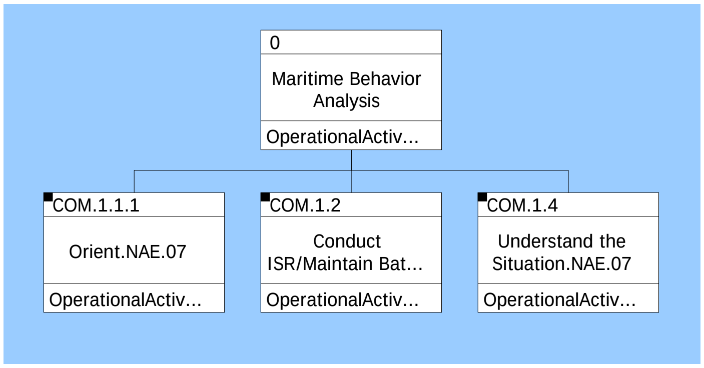


Figure 18. WMSE Operational Architecture (Tier 1)

The operational activities, which were discussed in the DRM, are in the first level of the hierarchy shown. These activities decompose into several more layers of abstraction, showing the several constituent activities, which make up the main activities. Figures 19 and 20 show the many sub-tier activities that were decomposed in the operational activity analysis, to ensure a thorough treatment of the mission scenario. Even with the large number of sub-tier activities that were derived, many of the common activities, which were listed in the COAL decompositions of the main and sub-tier activities were deleted from the architecture because they were not relevant, according to the particular parameters given in the DRM OPSIT. Often analysts will disregard the culling out of these extraneous activities embedded within the COAL decompositions, due to the tedious nature of the exercise. However, this trimming and refining of the operational architecture was an extremely important analytical step to ensure that the correct functions would be selected and mapped in the actual functional analysis and allocation steps.

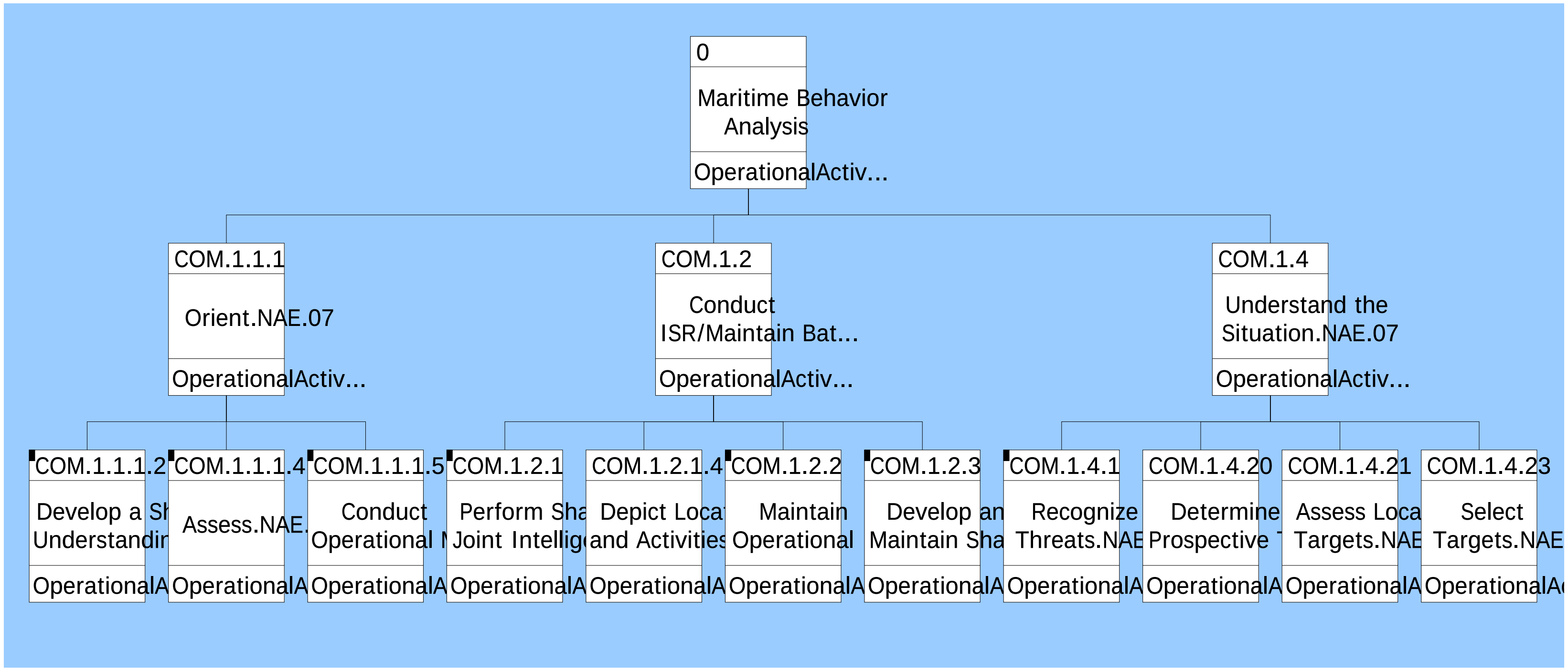


Figure 19. WMSE Operational Architecture (Tier 2)

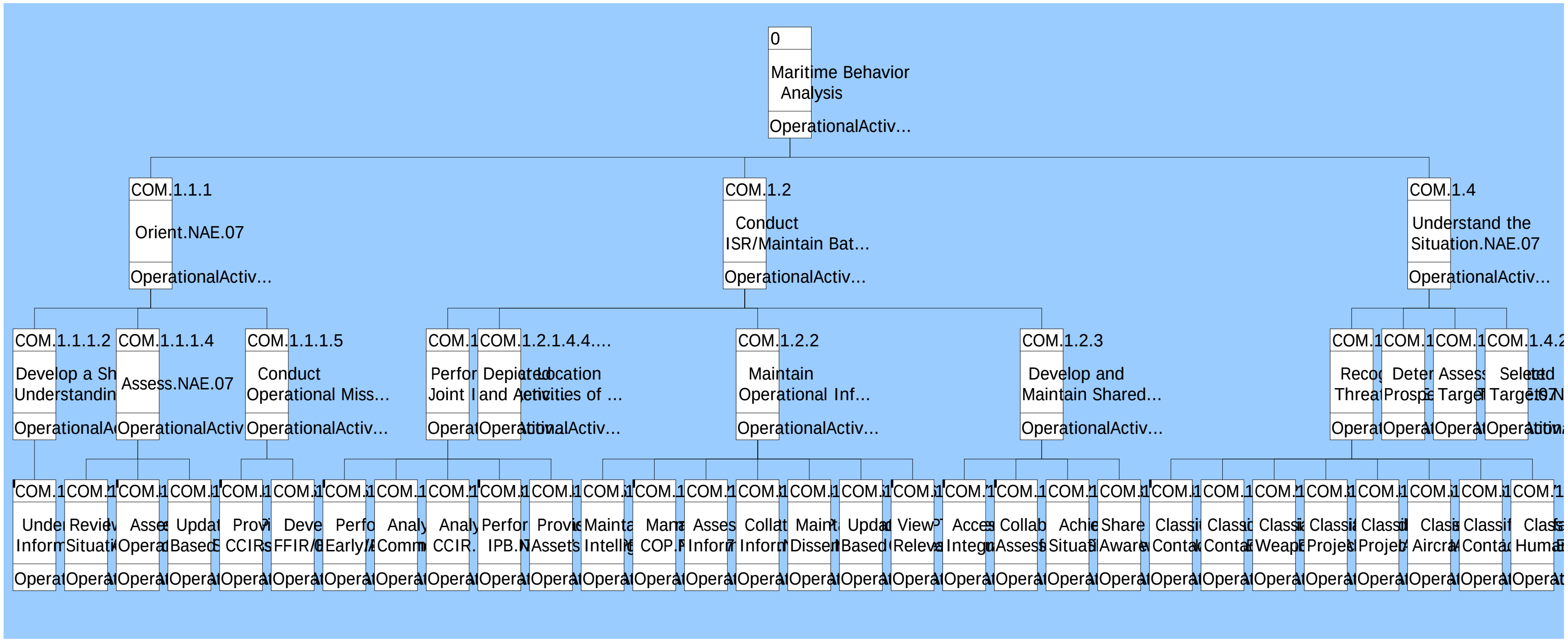


Figure 20. WMSE Operational Architecture (Tier 3)

It should be readily apparent how quickly untenable this becomes without a very large team of individuals working full time to analyze all the activities, map them to tasks, come up with the proper MOEs, and then run all the modeling simulations to test if the MOEs are even achievable within the architecture. Therefore, this architecture focuses on three primary tasks, which are decomposed by two main Sub-Activities, *Orient* and *Understand the Situation*. The first primary task, *Orient*, breaks down to *Assess* and then further to *Assess the Operational Situation* (COM.1.1.1.4.3). From this activity, the task *Analyze and Assess Situation* (NTA 5.2) from the DRM is executed. The MOE from this task, which seemed the most relevant and measurable from the list available, was *M1 – Minutes – To complete assessment of latest information (cycle time).*

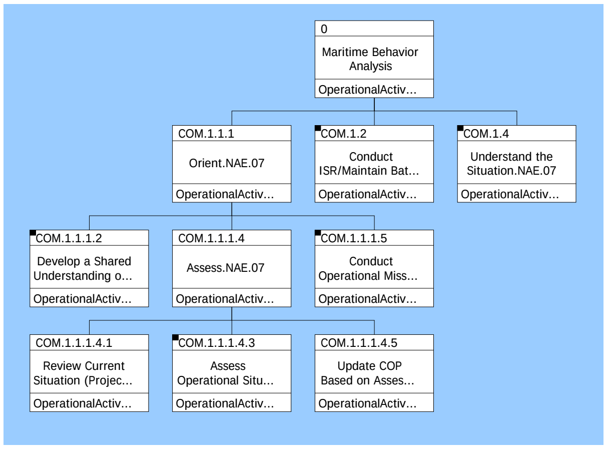


Figure 21. WMSE Assess OPSIT Decomposition

The next task is also decomposed from the *Assess* activity, but the sub-activity is the *Update COP based on Assessment* activity (COM.1.1.1.4.5), shown in Figure 21. This task is to *Provide Indications and Warning (I&W) of Threat* (NTA 2.4.5.3). Here, the MOE, which was relevant and measureable, is *M3 – Hours – Lead-time in predicting enemy*. The next task is decomposed under a different activity, *Understand the Situation* down to *Recognize Threats* and then finally down to the *Classify Surface Contacts (COM.1.4.1.7)* activity. This activity was assigned the task *Disseminate Tactical Warning Information and Attack Assessment (TA 2.4)*, from the UJTL. This task could easily have been applied to any or all of the sub-activities under Recognize Threats, however, although it is recognized that the BAM should have the capability to classify all the types of contacts listed, we are limiting the scope of the task measurement to surface contacts in keeping with the current CONOPS, as defined. The MOE for this task is: M1 – Minutes – To disseminate information. The M2 MOE turned out to be too subjective to accurately model, and was therefore deleted. The decomposition for this last task is shown in Figure 22:

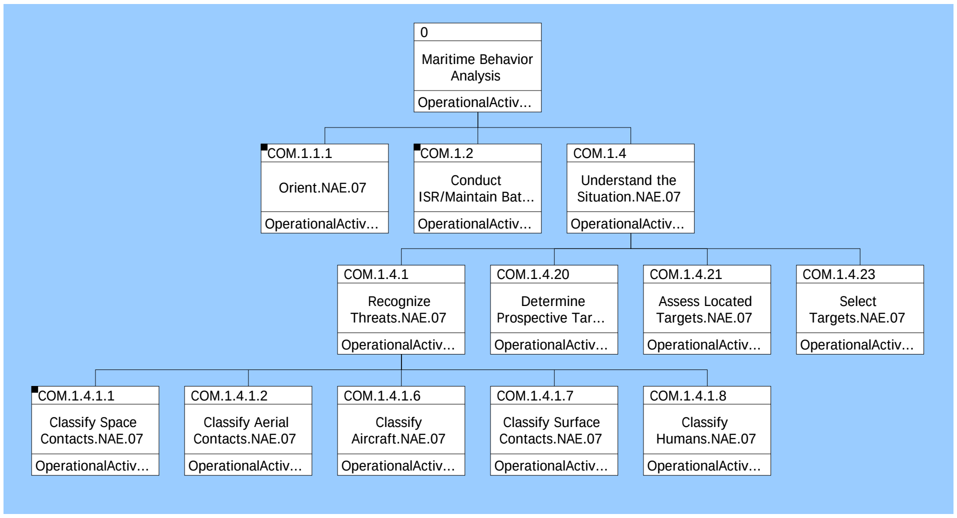


Figure 22. WMSE *Understand the Situation* (COM.1.4) Decomposition

2. Functional Architecture Definition

The process by which the functional architecture and hierarchy were built utilized generally the same type of methodology used in building the operational architecture, so those details will not be needlessly repeated here. The main differences in building the functional architecture were that the functions were derived by directly tying them to the Operational Activities that had the Operational Tasks to be accomplished and measured as described in the Operational Architecture described above. These three activities, again, were: *Assess the Operational Situation, Update COP based on Assessment* activity, and *Classify Surface Contacts (COM.1.4.1.7).*

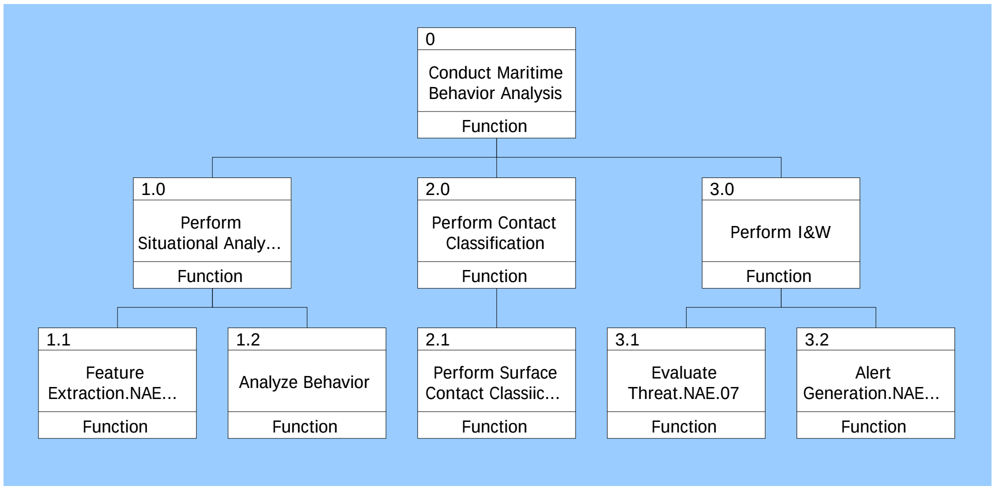


Figure 23. Functional Hierarchy – Tier 3

Figure 23 shows the hierarchy down to the third level of abstraction. Previously, these functions were broken down further, however, this was found to be unnecessary. In Figure 23, the three main functions are labeled 1.0, 2.0, and 3.0. These functions implement the Operational Activities listed previously to accomplish their respective Operational Tasks. Once the top-level functions were mapped to their respective activities, they were then decomposed into further sub-functions deemed necessary to accomplish the higher-level functions.

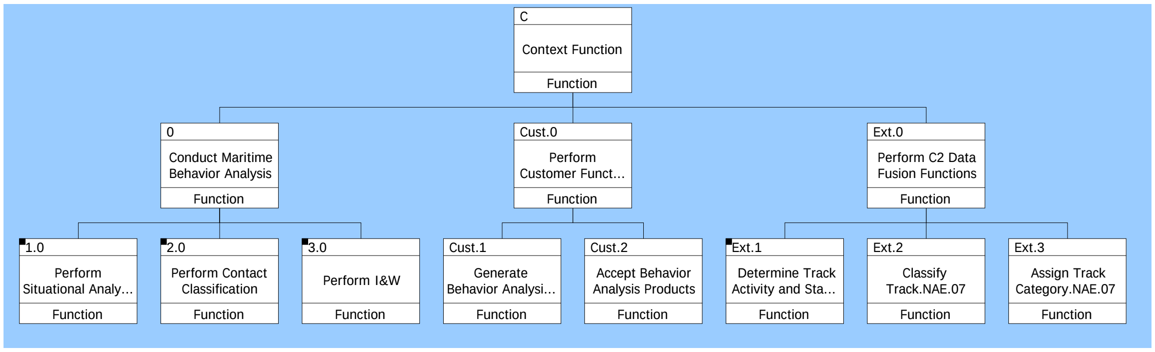

C
Context Function
Function
0
Conduct Maritime Behavior Analysis
Function
Cust.0
Perform Customer Funct...
Function
Ext.0
Perform C2 Data Fusion Functions
Function
1.0
Perform Situational Analy...
Function
2.0
Perform Contact Classification
Function
3.0
Perform I&W
Function
Cust.1
Generate Behavior Analysi...
Function
Cust.2
Accept Behavior Analysis Products
Function
Ext.1
Determine Track Activity and Sta...
Function
Ext.2
Classify Track.NAE.07
Function
Ext.3
Assign Track Category.NAE.07
Function

Figure 24. Functional Hierarchy – System Context

Once the primary sub-system functions were mapped to their respective operational activities, it became apparent that there needed to be a higher functional context in which these sub-functions would operate. Therefore, the system context function was created with external functions to the main MBAM function, such as customer functions and Command and Control (C^2)/Data Fusion functions. The external C^2 function provides the sensor coordination and data fusion for the input data, which must be provided to the MBAM. The Customer functions are vital to provide the context of generating requests for and acting as the recipients of the analyzed behavior data that the MBAM provides. It is this context and the interactions of these functions that allow the functional flow, shown in Figure 25, from which the functional behavior simulation is derived.

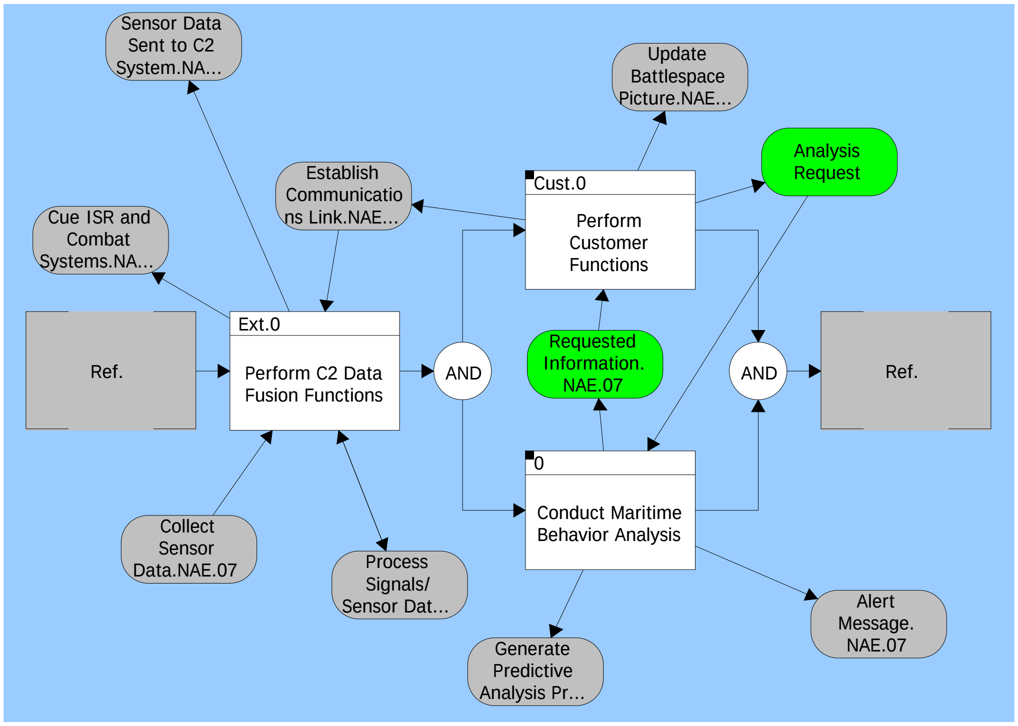

Sensor Data Sent to C2 System.NA...
Update Battlespace Picture.NAE...
Analysis Request
Establish Communicatio ns Link.NAE...
Cust.0
Perform Customer Functions
Cue ISR and Combat Systems.NA...
Ext.0
Perform C2 Data Fusion Functions
Ref.
AND
Requested Information. NAE.07
AND
Ref.
0
Conduct Maritime Behavior Analysis
Collect Sensor Data.NAE.07
Process Signals/ Sensor Dat...
Alert Message. NAE.07
Generate Predictive Analysis Pr...

Figure 25. WMSE System Context EFFBD

As seen in the functional flow block diagram (FFBD) in Figure 25, fused sensor data comes from the *Perform C^2/DataFusion* function and is then fed into the *Conduct Maritime Behavior Analysis* function. Simultaneously, the data coming into the MBAM function signals a trigger from the customer to initiate an Analysis Request. This request begins the process of the linear sub-functions within the MBAM parent function, shown in Figure 26.

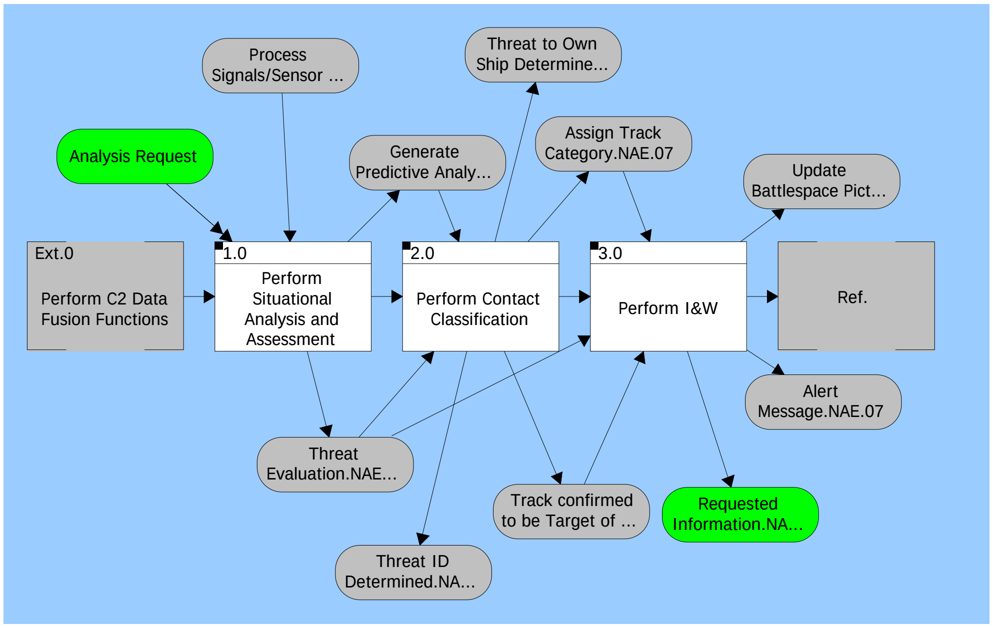

Process Signals/Sensor ...
Threat to Own Ship Determine...
Analysis Request
Generate Predictive Analy...
Assign Track Category.NAE.07
Update Battlespace Pict...
Ext.0
Perform C2 Data Fusion Functions
1.0
Perform Situational Analysis and Assessment
2.0
Perform Contact Classification
3.0
Perform I&W
Ref.
Alert Message.NAE.07
Threat Evaluation.NAE...
Track confirmed to be Target of ...
Requested Information.NA...
Threat ID Determined.NA...

Figure 26. MBAM Sub-functions

As these software-driven functions are executed serially, shown in the FFBD, they are "capturing" the available resource, which is processing power, according to certain random distributions of their expected usage of processing power. As shown in the simulation results, Figure 27, these functions, as they execute, will temporarily deplete the available processing capability, as well as take some discrete amount of time to complete. It is this usage of time (and in future research, usage of processing capacity) that can be compared against derived operational metrics to obtain the actual measures of effectiveness using this notional system architecture. Once the *Requested Information* item is outputted from the last function, shown in Figure 24, this triggers the customer function to then accept the behavior analysis product, which could be in the form of an alert message, a contact of interest report, or merely an "ops normal, no threat found" report. At this point in the sequence, the flow and the simulation end.

Figure 27 presents the graphical simulation output from CORE. The x-axis is time, which is depicted on the scale at the top of the simulation graph. The y-axis is processing capacity, which is depicted as a percentage of the total capacity available. Since the actual processing capacity, as well as processing requirements of a real-world system is unknown, notional values were derived that seemed to best represent a potential processing capability, as well as estimated processing requirements for the various functions. From the estimated value simulation shown, the time to complete a behavior analysis, from request to receipt of requested information is 74.97 seconds.

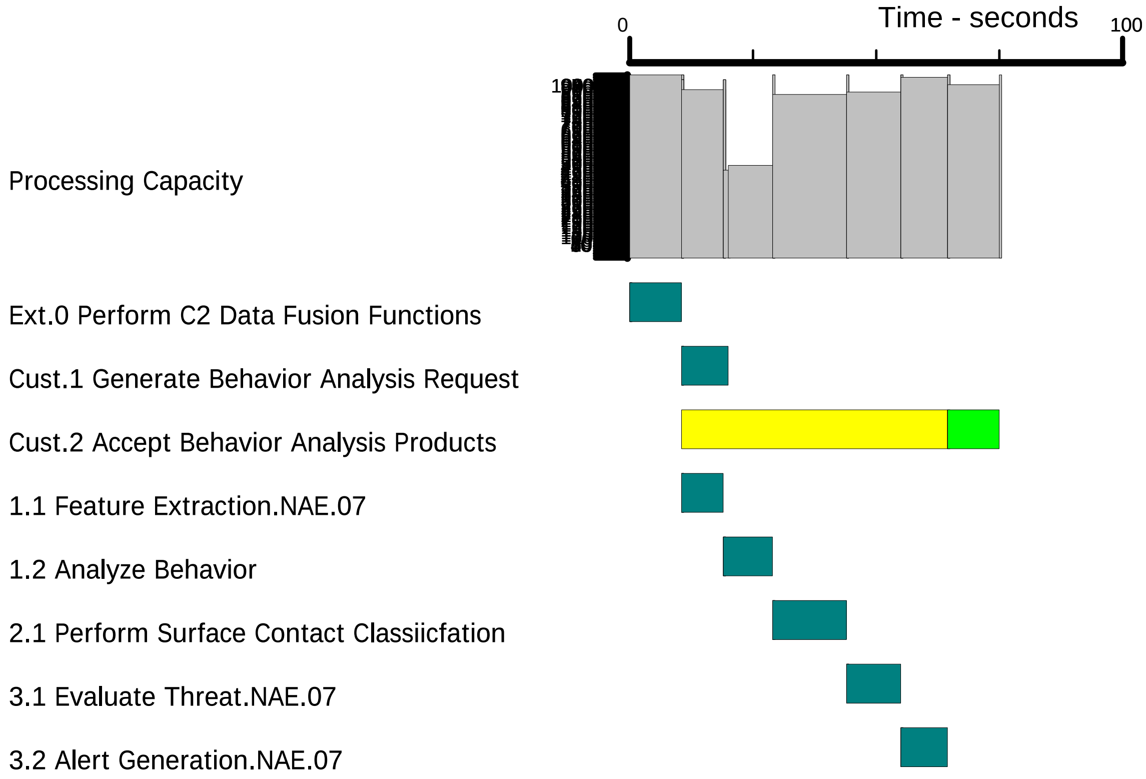


Figure 27. CORESim MBAM Simulation Run

To conclude, it took approximately 1 minute and 15 seconds to return a behavior analysis product to the user, and have it accepted, from the time the fused sensor data was sent to the MBAM. This value could be judged against the metrics of: Minutes – To complete assessment of latest information (cycle time); Hours – Lead-time in predicting enemy actions; and Minutes – To disseminate information. Future work may involve studies to determine which actual values for these metrics would be deemed acceptable to the user, which would certainly need to be accomplished for effective operational validation. However, from personal professional experience, a one-minute turn-around on a complex behavioral classification should be more than acceptable. The simulation at this time only has the capability to simulate the processing on one piece of data at a time.

Quite a significant amount, but not all of the processing capacity is used as the system is stressed. In order to understand the behavior with more accurate processing capability limits and usage rates for the various functions, as well as with multiple data items, the use of simultaneous and random discrete event queuing, as well as experimentally derived processing capacity and functional data processing values would be required in future research to more fully validate this model.

3. Requirements Architecture Definition

The requirements that were entered into the CORE application and mapped to the Operational Activities were pulled directly from the DRM. As with the previous development processes, the process of mapping the requirements directly to the defined activities caused reflection upon the initial requirements and therefore the requirement list and the DRM was refined even further. While pictures are said to be worth a thousand words, unfortunately, the format of the CORE hierarchical diagram does not make the following depictions in Figures 28 and 29 quite that useful, however, they do show at least the general concept of the mapping of the high-level requirements, which, as mentioned in the DRM, have been taken directly from the DoD’s MDA Joint Integrating Concept and identify Maritime Domain Awareness capability gaps

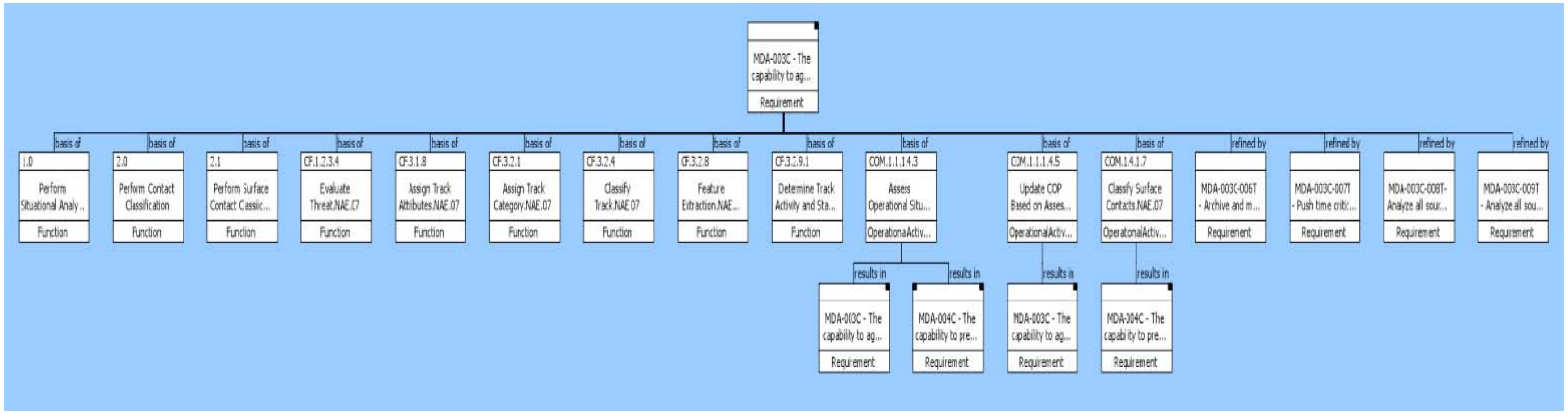

MDA-003C - The capability to ag...
Requirement
basis of
1.0
Perform Situational Analy..
Function
basis of
2.0
Perform Contact Classification
Function
basis of
2.1
Perform Surface Contact Cassic...
Function
basis of
CF.1.2.3.4
Evaluate Threat.NAE.07
Function
basis of
CF.3.1.8
Assign Track Attributes.NAE.07
Function
basis of
CF.3.2.1
Assign Track Category.NAE.07
Function
basis of
CF.3.2.4
Classify Track.NAE.07
Function
basis of
CF.3.2.8
Feature Extraction.NAE...
Function
basis of
CF.3.2.9.1
Detemine Track Activity and Sta...
Function
basis of
COM.1.1.14.3
Assess Operational Situ...
OperationaActiv...
basis of
COM.1.1.1.4.5
Update COP Based on Asses...
OperationalActiv...
basis of
COM.1.4.1.7
Classify Surface Contacts.NAE.07
OperatonalActiv...
refined by
MDA-003C-006T - Archive and m...
Requirement
refined by
MDA-003C-007T - Push time critic...
Requirement
refined by
MDA-003C-008T- Analyze all sour...
Requirement
refined by
MDA-003C-009T - Analyze all sou...
Requirement
results in
MDA-003C - The capability to ag...
Requirement
results in
MDA-004C - The capability to pre...
Requirement
results in
MDA-003C - The capability to ag...
Requirement
results in
MDA-004C - The capability to pre...
Requirement

Figure 28. MDA-003C –"The capability to aggregate, display, and analyze maritime information in order to understand the maritime environment and identify threats" relationships

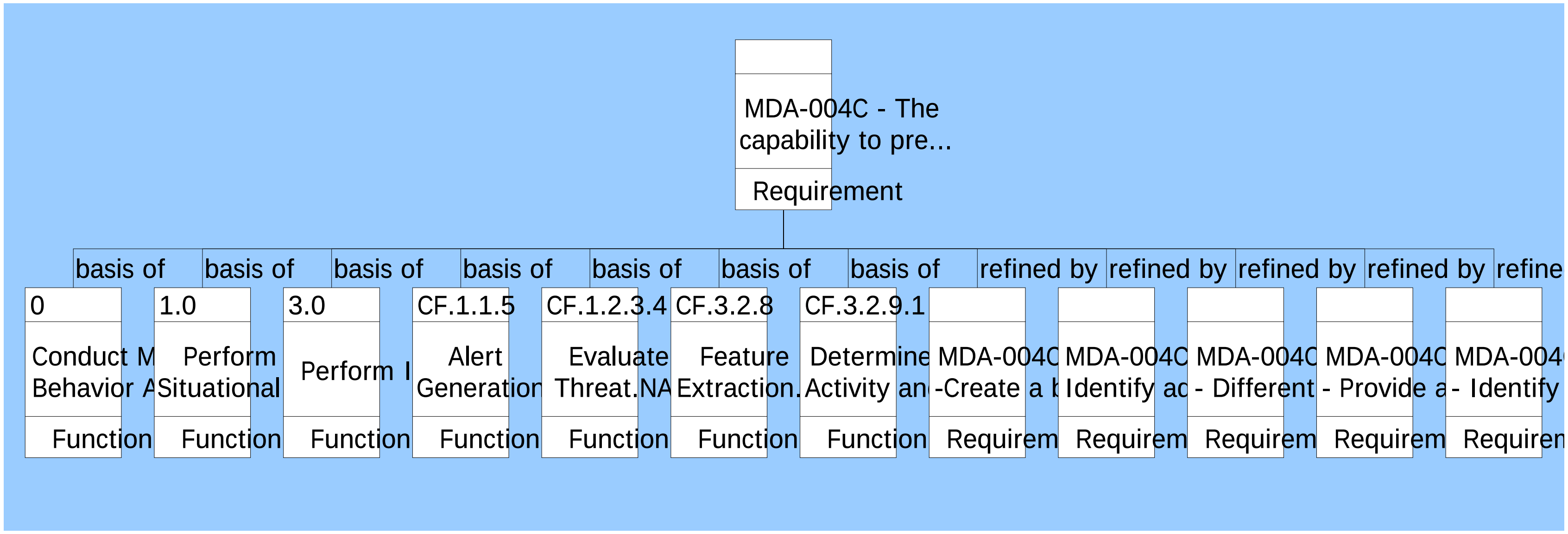


Figure 29. MDA-004C –"The capability to predict activity within the maritime domain" relationships

Figures 28 and 29 illustrate the many relationships that the requirements have with Functions, Operational Activities, and other refined sub-tier requirements.

All of the requirements defined were considered to be originating requirements, as they originated from outside the architecture or systems development process (as opposed to derived requirements, which emanate from within these processes). Additionally, these requirements are considered functional requirements, as they deal with those elements, which make the system function in order to meet the required capabilities. Non-functional requirements were considered, such as reliability, maintainability, availability, and the like, however, not only were those requirements not mentioned in the requirement source document, the MDA JIC, but as it was observed that at this early stage in the process those non-functional requirements are not yet known, and therefore could not be defined and mapped at this time.

The Measures of Performance (MOP's) and Technical Performance Measures (TPM's) are derived from the MOE's defined earlier in Paragraph 1. Those MOE's, as described before, are:

- M1 – Minutes – To complete assessment of latest information (cycle time)
- M3 – Hours – Lead-time in predicting enemy actions
- M1 – Minutes – To disseminate information

Some initial possible TPM's listed below are derived from the translation of the MOE's into the functional model simulation described in the previous section.

- Data speed into and out of the MBAM (in terms of a baud rate, or amount of data per second that can be uploaded to and downloaded from the MBAM)
- Processing speed (in terms of MHz or GHz or seconds) of the MBAM hardware
- Processing capacity (in terms of MBps or GBps) of the MBAM hardware
- Processing efficiency (in terms of MBps or GBps) of the MBAM software

- Speed of alert data generation (several different types of measures could be applied here, from time from initial contact notification to alert reception by external element outside the BAM, to time of hostile intent classification to alert generation and transmittal)

Some additional notional TPMs that are being considered regarding behavior analysis efficiency and accuracy are:

- Number of contacts processed within a certain time period
- Number of contacts whose behavior was correctly identified
- Number of false alerts out of the total number of alerts (error rate)

It is anticipated that with improvement and expansion of the model in future recommended research, as well as with the integration and testing of various new software components to validate and improve the behavior models, the MOPs and TPMs will be able to be more clearly defined.

4. System (Physical) Architecture Synthesis

Given that the functions accomplished by the Maritime Behavior Analysis Module (MBAM) will be via software, the only components that made sense to evaluate were the hardware components that the software would need to run on in order to both execute the MBAM functions. In addition, this hardware must have network interfaces to be able to pull data in to the algorithm to execute the functions, as well as send data out once the functions are complete. Therefore, the components analyzed were the basic components of a standard COTS desktop computer, which, fundamentally, would have the processing power and capability to run the MBAM software and interface with a standard Ethernet-based or wireless network.

Different types of components which would be needed to process the algorithms, in addition to components that would need to interface with input and output external networks were analyzed, comparing the various given types and performance features of those components. Those features were organized into a morphological matrix, shown in Table 6.

Feature	**Option 1**	**Option 2**	**Option 3**
Processor Speed	1.6 Ghz	2.4 Ghz	3.2 Ghz
Bus Speed	400 Mhz	800 Mhz	
RAM Speed	400 Mhz	667 Mhz	
Cache Memory	1 MB	2 MB	4 MB
Hard Drive Capacity	250 GB	320 GB	500 GB
Networking Type	Bluetooth	Integrated 10/100/1000 Network Card	
Data Link Protocol	Gigabit Ethernet	IEEE 802.3u	

Table 6. Morphological Matrix of Options

Assuming cost differences in COTS computer components of varying levels of performance are roughly negligible, the components with the highest values of the first five features, which dealt with processing performance, were chosen as the most desirable. Upon this decision, the remainder of the features, which dealt with networking performance and compatibility, was grouped into 4 possible permutations.

1. Max performance – Bluetooth – Gig Ethernet
2. Max performance – Bluetooth – 802.3u
3. Max performance – Integrated Network Card – Gig Ethernet
4. Max performance – Integrated Network Card – 802.3u

The permutation options were then compared using a Pugh selection matrix, shown in Table 7. The first option was chosen as the comparison datum, and then each other option was compared to it, based on certain performance criteria. These criteria were selected by consulting with operational SMEs as to the OMOEs that would be evaluated for these types of components and their associated functions.

Criteria	**DATUM (Opt 1)**	**Option 2**	**Option 3**	**Option 4**
Processing Speed	D	S	S	S
Processing Power	A	S	S	S
Network Interface Speed	T	-	+	+
Network Interface Compatibility	U	-	+	+
Network Interface Reliability	M	-	S	+
SUMS OF POSITIVES		0	2	3
SUMS OF SAMES		2	3	2
SUMS OF NEGATIVES		3	0	0

Table 7. Pugh Selection Matrix

Based upon the comparison analysis above, it appears that Option 4, the component set with the maximum processing speed and power, combined with an Integrated 10/100/1000 Ethernet Network Card and a IEEE 802.3u Network protocol would provide the fastest, most stable, and compatible network interface for the MBAM to operate most effectively within the overall system.

Once the component concept generation and selection was complete, the building of the physical architecture could begin. This component selection and build of the system from the physical architecture is discussed in the Network Topology section of the WMSE Proof of Concept System description in Chapter V. As with the functional architecture, the physical architecture required components in a higher system context in which to operate and map back to the functions that they perform. Figure 30 shows the WMSE System Context Component Architecture.

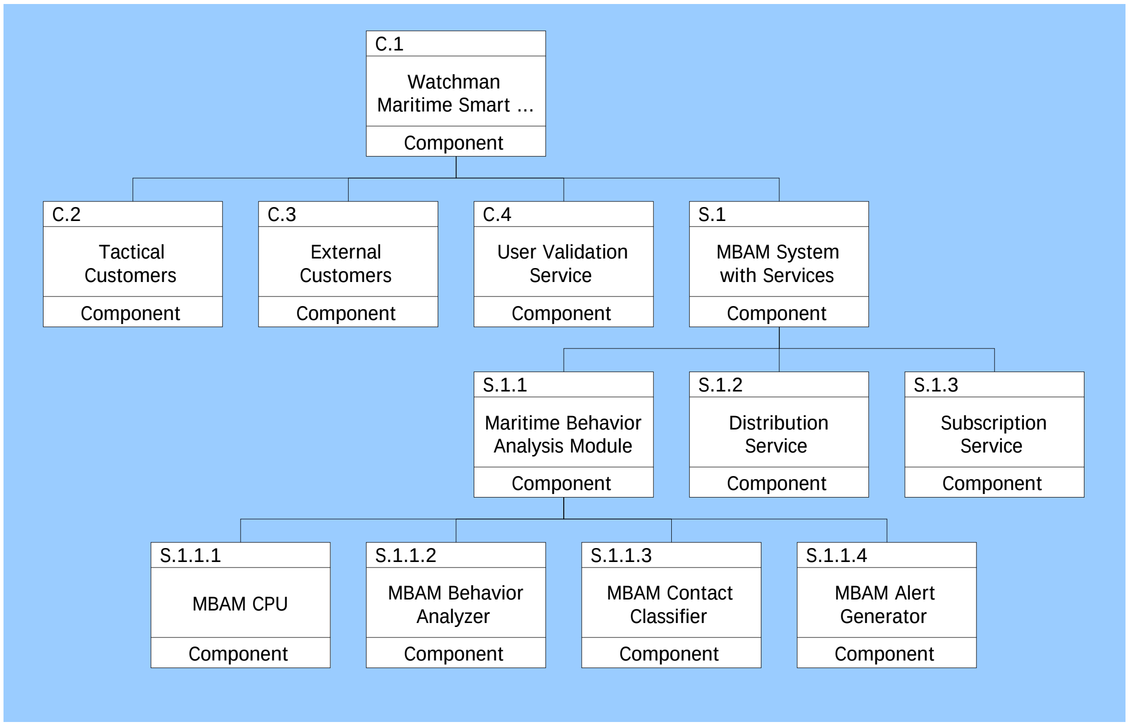


Figure 30. WMSE System Context Component Architecture

Not every element of this architecture will be explained; however, beginning with the first tier, the components include those customer and external components that perform the associated functions of passing data to and from the MBAM, as well as validating that data, as part of the greater WMSE system. The next level down decomposes component S.1, which is the MBAM System "with services." These services are not modeled or included in the functional architecture fully as yet, but were included as future growth items, assuming that the MBAM data would be desirable to external users that would need to subscribe to and be distributed MBAM data analysis services. Finally, the "meat" of the architecture is decomposed as the MBAM itself, with its associated hardware (as described in the component selection above) and software "modules" that perform the individual MBAM functions, which were modeled in the EFFBD and simulation.

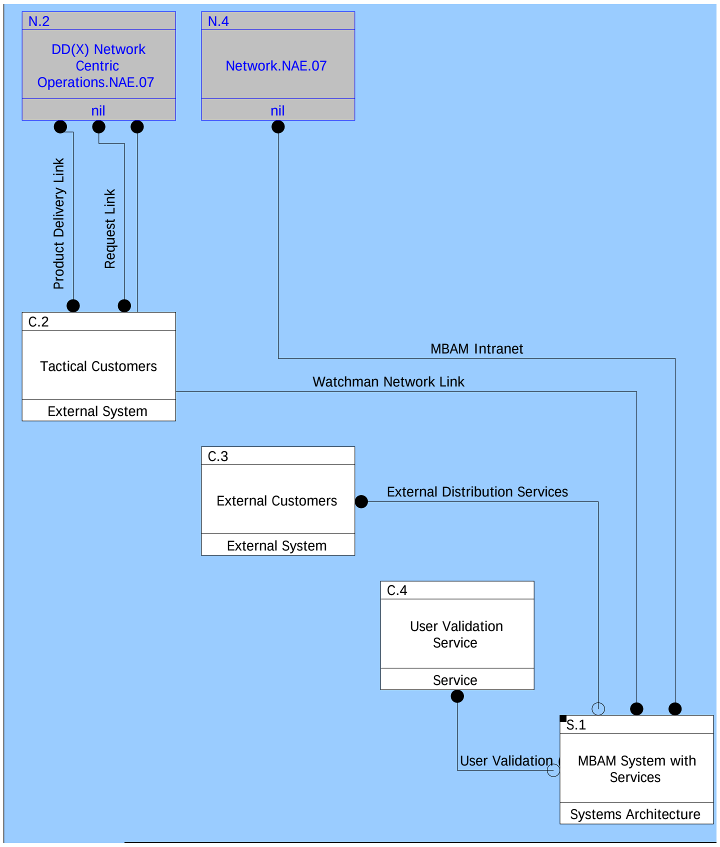


Figure 31. WMSE System Context Block Diagram

Just as the system functions had information linkages, these components have physical linkages, which are depicted via various information links. The relationships of the components to each other in the physical architecture by these links are shown in Figures 31–32.

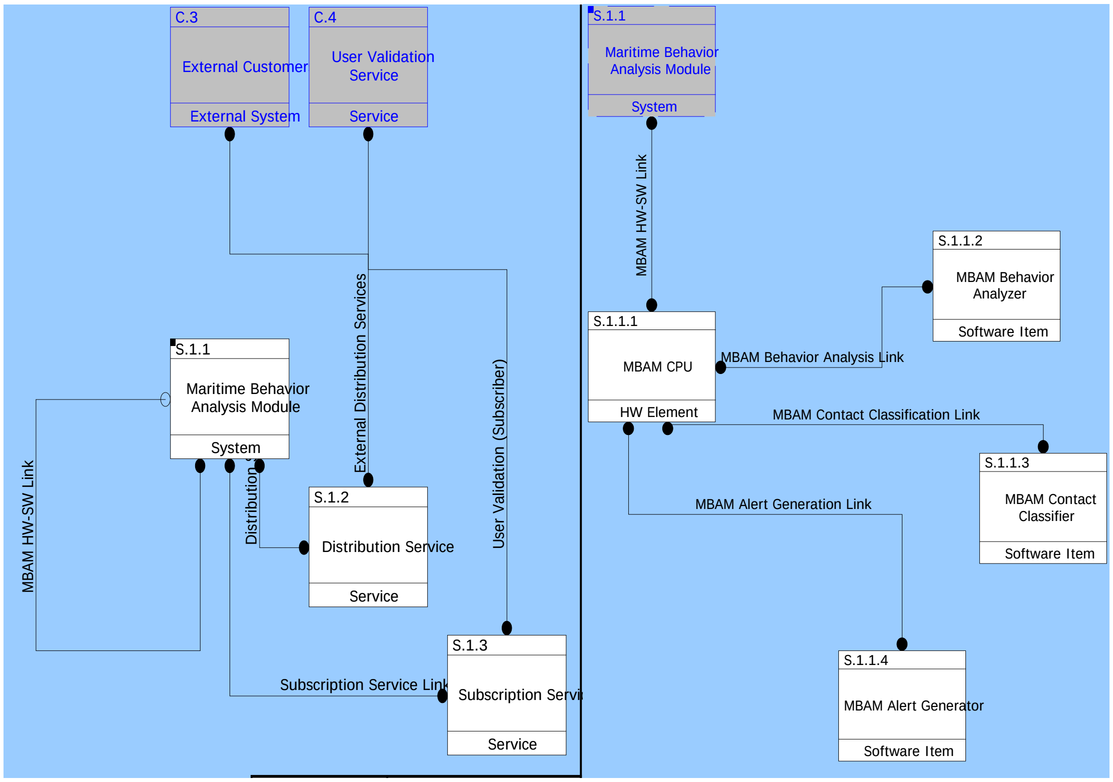


Figure 32. MBAM System with Services Block Diagrams

5. Initial Architecture Development Conclusion

These functional and physical architectures, along with their respective information and information-carrying links were the final step in building the framework upon which the initial modeling of the speed and processing effectiveness of this system segment was accomplished. This work has laid the foundation for what is to come in any follow-on projects, which will further refine the system requirements and identify any operational or capability shortfalls that exist within the scope of the defined mission and architecture.

The next phase encompassed the implementation of the functional model for the *Conduct Maritime Behavior Analysis* function by the actual demonstration of software elements performing those functions in a laboratory environment. This was accomplished with the ultimate goal being met of integrating those functions and components into a high-level functional laboratory system, demonstrating the capability and feasibility of an entire Smart Sensor Network system. In addition, the next section shows how the WMSE project team integrated the individual project pieces into a complete system operational and functional architecture, which was then demonstrated to validate the design of the entire system.

B. INTEGRATION ARCHITECTURE

The next step in the systems architecting process was to take the initial architecture that was developed and further refine it for integration into an actual, functioning system. This integration architecture shows the refinements and improvements which were made to the initial architecture, as a result of the architecture design and development process, as well as the inclusion of the other main elements of the WMSE system, comprising all of the Detect, Track, Classify, and Respond high-level functions. These high-level functions correspond to the DIPR system construct, discussed in Chapter III.

The integration architecture of the system was also completed using ViTech's CORE software by entering and mapping the system's architecture elements, including the process and functional decomposition into the program for analysis and documentation. Process, functional and component relationships within the system architecture, along with enhanced functional flow block diagrams (EFFBD), IDEF0, and N2 diagrams depicting these relationships, are provided by the System Description Document (SDD) report from CORE. The SDD is included as an Appendix of this report and includes detailed description and illustration of the system's abstraction, coupling and cohesion.

The functional and process decompositions to the first level of abstraction are shown in the following diagrams. Identifying and decomposing processes, as well as mapping them to functions are critical for integration. While functions relate to "what" must be done, processes reveal the "how" the functions are accomplished in a system. Each function must have a process; however, extraneous processes must be identified and eliminated from the architecture to ensure a robust and efficient integration design. Further detail of the lower level abstractions can be found in Section 7 of the SDD.

Process Number & Name	Function Number & Name
0 Domain Awareness Enhancement Process	0 Enhance Domain Awareness
1 Contact Detection (WiLife) Activities	1.0 Detect
1.1 Assign Camera Settings Process	1.5 Manage Video Collection
1.2 Domain Monitoring Process	1.1 Monitor Domain
1.2 Node Command Center Activities	O Perform Overhead Functions
1.3 Motion Detection Process	1.2 Determine Contact Presence
1.4 Video Record Process	1.3 Collect Video Data
1.5 Video File Write Process	1.4 Store Video Data
2 Contact Tracking (Blob Tracker) Activities	2.0 Track
2.1 Read Video File Process	2.1 Read Video Data
2.2 Process Video File Process	2.2 Process Video Data 2.3 Build Contact Track
3 Classification Activities	3.0 Classify (Conduct Behavior Analysis)
3.1 Server Activities	
3.1.1 Database Activities	
3.1.1.1 DB Read Process	3.1 Read Contact Data Files
3.1.1.1.1 DB Query Process	
3.1.1.2 DB Write Process	
3.1.1.2.1 DB Add Process	
3.1.1.2.2 DB Update/Modify Process	
3.1.2 GUI Activities	3.4 Perform I&W
3.1.2.1 GUI Command Process	O.3 Accept Mission Plan
3.1.2.1.1 GUI Push Process	
3.1.2.1.2 GUI Pull Process	
3.1.2.1.2.1 GUI Query Process	
3.1.2.2 GUI Display Process	
3.1.2.3 GUI Alert Process	3.4.3 Alert Generation.NAE.07
3.1.2.3.1 Send React Order (Abnormal Beh)	
3.1.2.3.2 Send React Request to User (Unknown Beh)	
3.1.3 BAM Activities	3.2 Analyze Behavior
3.1.3.1 BAM Pull Data	3.2.1 Read Observed Sequence 3.2.2 Read Stored Behavior Sequences
3.1.3.2 BAM Analyze	3.3 Perform Contact Classification
3.1.3.2.1 Cost Matrix Creation	3.2.4 Build Cost Matrix
3.1.3.2.2 Production Matrix Creation	3.2.3 Build Production Matrix
3.1.3.2.2.1 Process Stored Behavior Sequences	
3.1.3.2.2.2 Process Observed Sequence	

Process Number & Name	Function Number & Name
3.1.3.2.2.3 Parse ObsSeq into BehSeqs via State Machine	
3.1.3.2.3 Cost Calculation	3.2.5 Calculate Costs
3.1.3.3 BAM Determine Class	3.3.1 Classify Normal 3.3.2 Classify Abnormal 3.3.3 Classify Unknown
3.2 User Activities	O.4 Perform User Interface Functions
3.2.1 System Initialization	O.1 Perform Startup O.2 Perform Vehicle Subsystem BIT
3.2.2 GUI Interface	3.4.2 Evaluate Threat.NAE.07
3.2.2.1 Monitor GUI	
3.2.2.2 Evaluate Alerts	3.4.2 Evaluate Threat.NAE.07
3.2.2.3 Alert Response	
3.2.2.3.1 Threat Dismissal	
3.2.2.3.2 Threat Affirmation	
4.0 Response	4.0 Respond
4.1 WSSN C^2 Activities	
4.1.1 WSSN Tasking	
4.1.1.1 WSSN Tasking Order Data Receipt	4.1.1 Receive WSSN Tasking Order Data from Watchman Server
4.1.1.2 WSSN Tasking Order Generation	4.1.2 Generate WSSN Tasking Order
4.1.1.3 WSSN Tasking Order Transmission	4.1.3 Send WSSN Tasking Order to WSSN
4.1.2 WSSN Tasking Response	
4.1.2.1 WSSN Tasking Order Response Receipt	4.2.1 Receive WSSN Tasking Order from WSSN C^2 4.4.3 Receive WSSN Tasking Order Response from WSSN
4.1.2.2 WSSN Tasking Order Response Processing	4.4.4 Process WSSN Tasking Order Response
4.2 WSSN Agent Activities	
4.2.1 Initialization	O.6 Initialize WSSN Agent
4.2.1.1 Diagnostics	O.6.1.1 Perform Diagnostics
4.2.1.2 Primary Task Designation	O.6.1.2 Designate Primary Tasking
4.2.1.3 WSSN Tasking Order Receive Mode	O.6.3 Enter WSSN Tasking Order Receive Mode
4.2.2 WSSN Tasking Order Processing	
4.2.2.1 WSSN Tasking Order Receipt	
4.2.2.2 Task Agent Determination	4.2.3 Determine Task Agent
4.2.2.2.1 Own Battery Voltage Determination	4.2.2.1 Determine Own Battery Voltage
4.2.2.2.2 Own Battery Voltage Transmission to Other Agent	4.2.2.2 Send Own Battery Voltage to Other Agent

Process Number & Name	Function Number & Name
4.2.2.2.3 Battery Voltage Receipt from Other Agent	4.2.2.3 Receive Battery Voltage from Other Agent
4.2.2.2.4 Battery Voltage Comparison	4.2.3.4 Compare Battery Voltages
4.2.3 WSSN Tasking Order Execution	
4.2.3.1 Positioning for COI Data Collection	
4.2.3.1.1 Navigation to COI Zone	
4.2.3.1.2 Maneuvering to COI	4.3.3 Maneuver to COI
4.2.3.1.3 COI Detection	4.3.2 Detect COI
4.2.3.2 COI Data Collection	
4.2.3.2.1 COI Image Collection	4.3.4.1 Collect COI Image
4.2.3.2.2 COI Color Collection	4.3.4.2 Collect COI Color
4.2.3.3 WSSN Tasking Order Response	4.4 Respond to WSSN Tasking Order
4.2.3.3.1 WSSN Task Order Response Generation	4.4.1 Generate WSSN Tasking Order Response
4.2.3.3.2 WSSN Tasking Order Response Transmission	4.4.2 Send WSSN Tasking Order Response to WSSN C^2

Table 8. Process to Function Allocation Mapping

In addition, Table 8 shows the relationships of the functions to the processes. As illustrated, not every process is mapped to a function, as there are several subsidiary processes that may be decomposed from a higher-level process. It is the higher-level process, in many cases, that is accomplishing the function, supported by the lower level process. This mapping of relationships between functions and processes, accomplished through an analysis of process and function abstractions, revealed both redundant and overlapping processes, as well as missing processes that had to be accounted for in the design. This allocation process of the integration architecture phase led to an improvement of the integration across the entire system, in addition to a much more streamlined and effective integration of the WSSN (as an agent-based representation of the *React* DIPR subsystem) into the WMSE POCS.

1. Functional Decomposition Hierarchies

This section presents the first level decompositions of the main functions in the WMSE POCS system functional architecture. These decompositions show how the main functions are allocated and broken down into their constituent sub-functions, ensuring that the higher-level functions can be accomplished in a reasonable fashion. The first level only is depicted here to give the reader a sense of the decomposition process. The full architecture, showing all the decompositions down to their lowest level of abstraction (the system and subsystem execution level) is given in the WMSE POCS description document, in the Appendix.

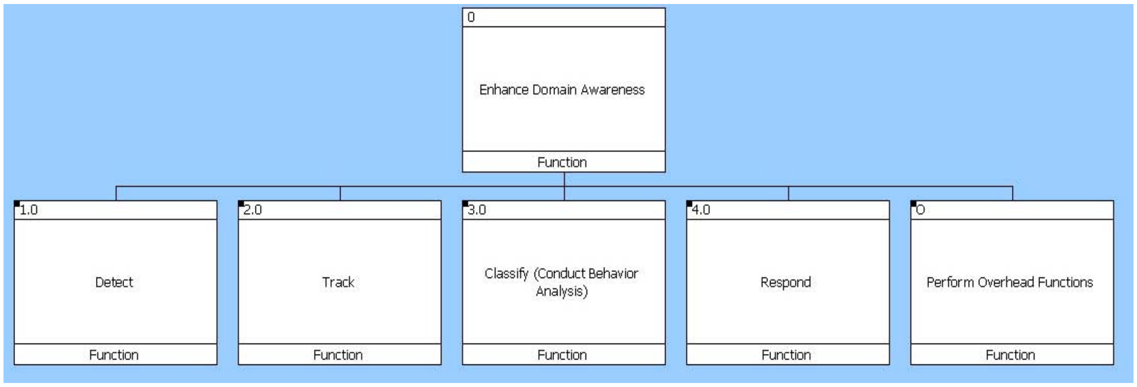


Figure 33. Enhance Domain Awareness Function – Level 1

Figure 33 shows the system context, or main, function, which is Enhance Domain Awareness, broken down into the four primary system functions, Detect, Track, Classify, and Respond, as well as a Perform Overhead Functions function. This last function is distinct from the primary functions in that the primary functions are related to operational activities, but Perform Overhead Functions is necessary for system operation by accounting for such functions as startup, shutdown, built-in-test (BIT), etc. The primary functions relate directly to the DIPR subsystems necessary for an AI system, and are mapped to the operational activities derived from the mission, as described in the Operational Architecture section. The main function is related to the overarching operational activity, which is enhancing maritime domain awareness, according to the DRM.

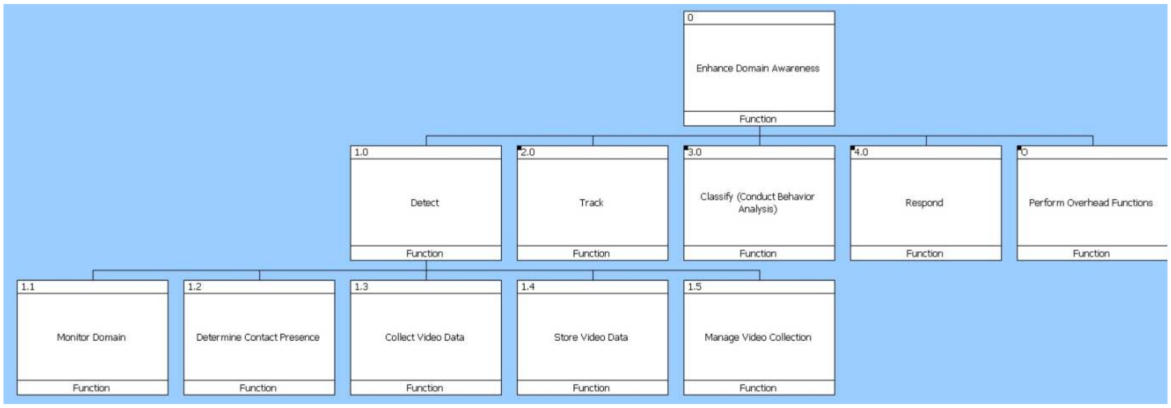


Figure 34. Detect Function – Level 1

Figure 34 depicts the first level decomposition of the first primary function, which is Detect. This, intuitively, represents the *Detect* DIPR subsystem, and its decomposed sub-functions show the types of functionality required to accomplish detecting a target.

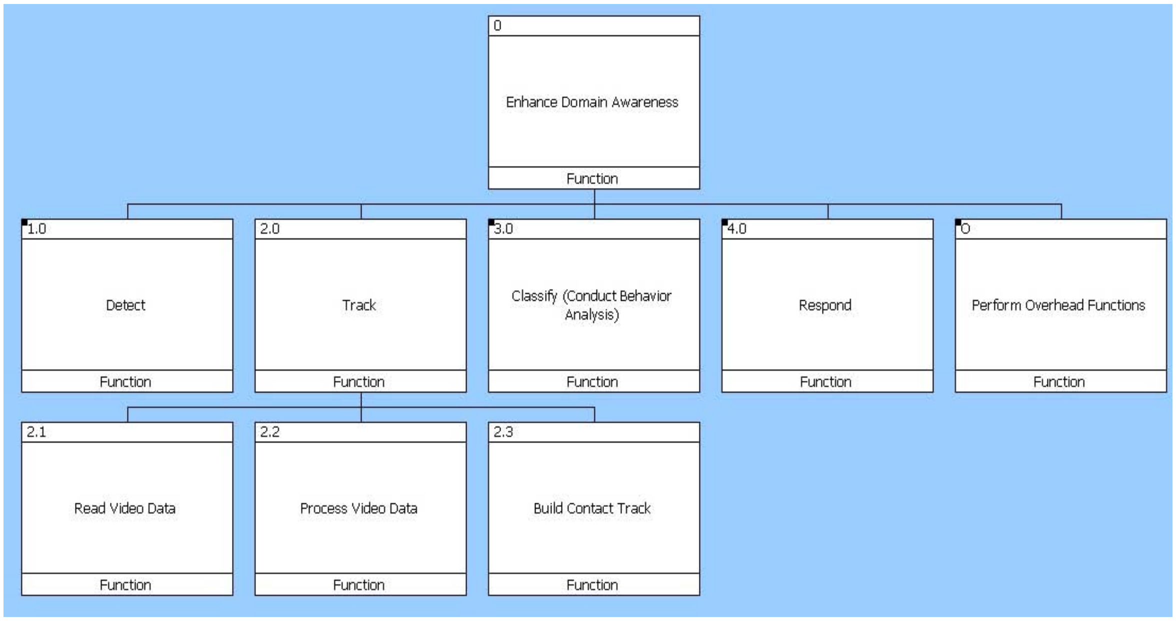


Figure 35. Track Function – Level 1

Figure 35, as with the previous figure, depicts the decomposition of the sub-functions that were determined to be required to accomplish the Track primary function. This function would also represent the *Identify* DIPR subsystem, as this function would conduct the feature fusion and intelligent state (symbol) creation required for automation, through the Process Video Data sub-function. Tracks are derived from fusing different types of features detected within the *Detect* subsystem. Additionally, functions that perform more specific feature extraction would have to be superimposed upon the Process Video Data sub-function to extract features and perform the transformation of those features into intelligent states (symbols) for use by the Classify function, Figure 36.

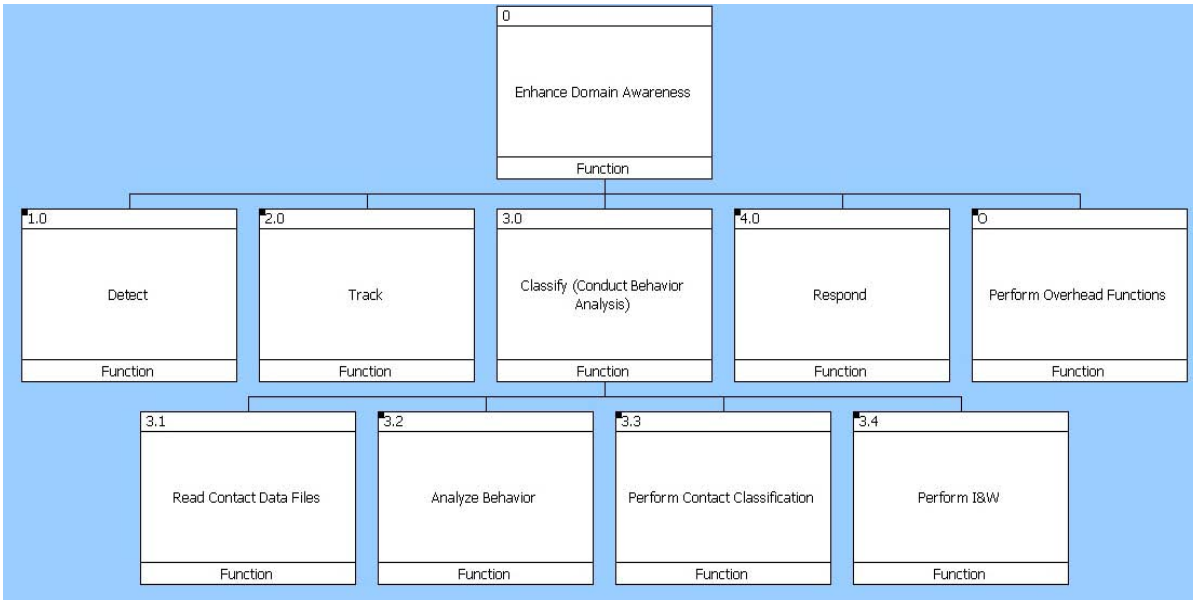


Figure 36. Classify Function – Level 1

Figure 36 shows the sub-functions required to perform the functions of a *Predict* DIPR subsystem. Sub-functions 3.2 and 3.3 correspond to the functions performed in the Behavior Classifier Module discussed in the first section of Chapter III, and are critical for automated behavior analytical processing. Sub-function 3.4 corresponds to the Infer Prediction Outcomes Module of the *Predict* model and is the last function in sequence that then passes the classification data to the next primary function Respond.

The Respond function, with its constituent sub-functions, is shown in Figure 37. This function, relating to the *React* subsystem of DIPR, utilizes the functions of the integrates WSSN to then perform other sub-functions required to react to the output of the Classify function, even if that reaction is to do nothing and continue monitoring the contact of interest (COI) and the domain.

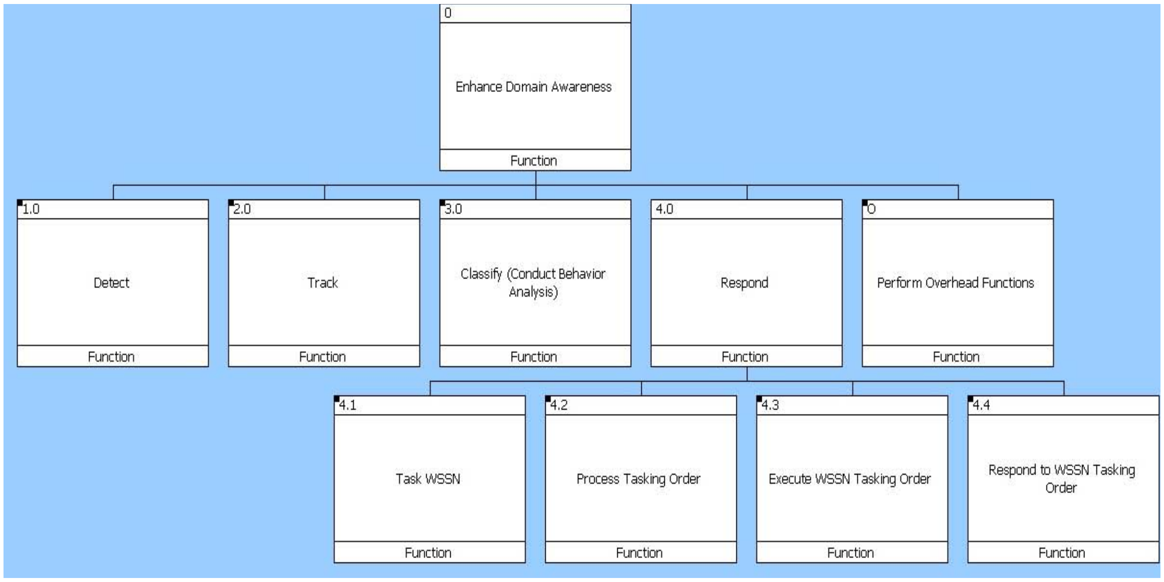


Figure 37. Respond Function – Level 1

The last function depicted in this section is the Perform Overhead Functions function, shown in Figure 38. As mentioned previously, this function does not relate to any particular subsystem of DIPR, but its sub-functions are very important to the routine operation and maintenance, the "housekeeping" of the system. These sub-functions include starting up and shutting down the system, performing system BIT, providing a user interface to the system, and allowing the user to input a customizable mission plan.

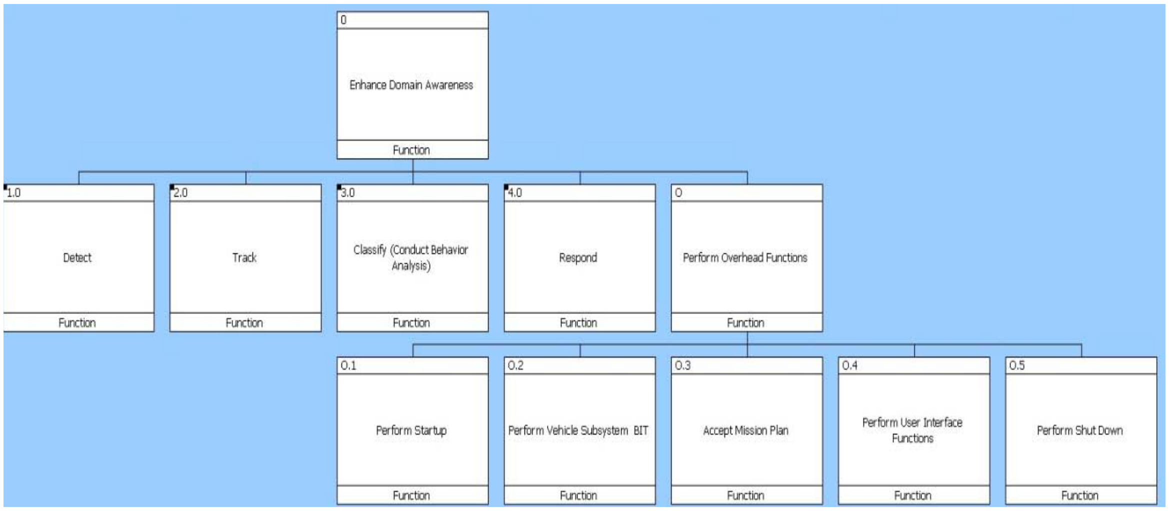


Figure 38. Overhead Function – Level 1

2. Process Decomposition Hierarchies

Quite similar to the previous section, this section presents the first level decompositions of the main processes in the WMSE POCS system process architecture. These decompositions show how the main system processes are allocated and broken down into their constituent sub-processes, ensuring that the higher-level processes can be accomplished in a sound manner. The first level only is depicted to describe the decomposition process. The full architecture, showing all the decompositions down to their lowest level of abstraction (the system and subsystem execution level) is given in the WMSE POCS description document, in the Appendix.

The processes shown and that are further decomposed in the SDD do not stand alone, but are mapped directly to the system functions described previously. This mapping is shown in Table 8. The processes are identified and decomposed in the system architecture, mapping to functions, to illustrate those means by which the functions of the system are actually accomplished. This process analysis and allocation methodology greatly assists the systems engineer in understanding what processes, which often are derived from the operational activities, need to occur in order to achieve a particular functionality. By understanding what processes or activities need to occur, as well as understanding what processes may be redundant or be applied across more than one function, the system can be more efficiently designed and integrated. This efficiency is realized as the processes are allocated across system components and elements to perform system functions. Additionally, as the breakdown of processes is understood across system components, system and component interfaces can be much more completely and effectively accounted for in the design.

An important note to the reader:

The process diagrams in this section were created from the CORE architecture development software tool. The tool is limited in that it does not include a specific category for “Process” elements. Therefore, the only reasonable element category that would adequately decompose and depict process elements, as well as properly map them to functional elements were Operational Activities. Thus, although the diagram blocks with the process names are each labeled “OperationalActivity,” they are indeed processes, and should not be confused with the actual operational activities of the operational domain, which comprise the Operational Architecture, which was discussed previously.

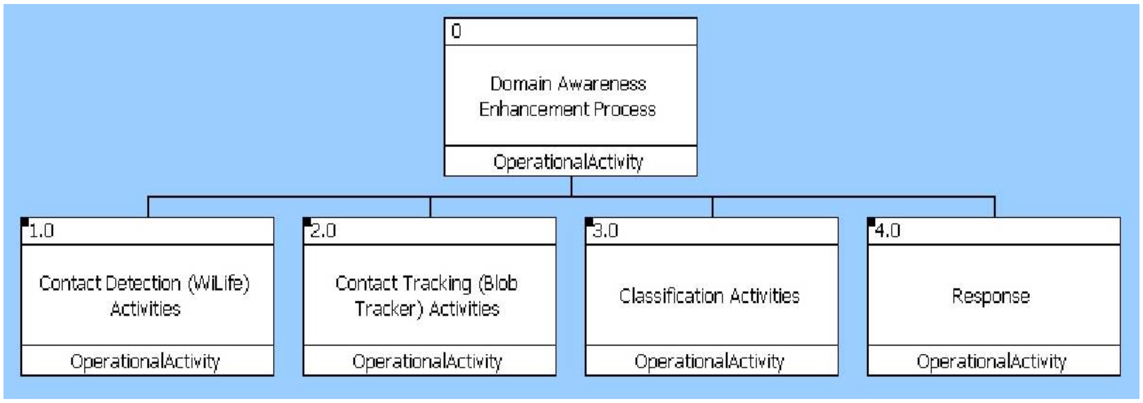

0
Domain Awareness Enhancement Process
OperationalActivity
1.0
Contact Detection (WiLife) Activities
OperationalActivity
2.0
Contact Tracking (Blob Tracker) Activities
OperationalActivity
3.0
Classification Activities
OperationalActivity
4.0
Response
OperationalActivity

Figure 39. Domain Awareness Enhancement Process – Level 1

Figure 39 shows the system context, or main, process, which is Enhance Domain Awareness Process, broken down into the four primary system processes, or activities, Contact Detection, Contact Tracking, Classification, and Response. These primary processes map directly to the primary functions in the first level of the functional hierarchy.

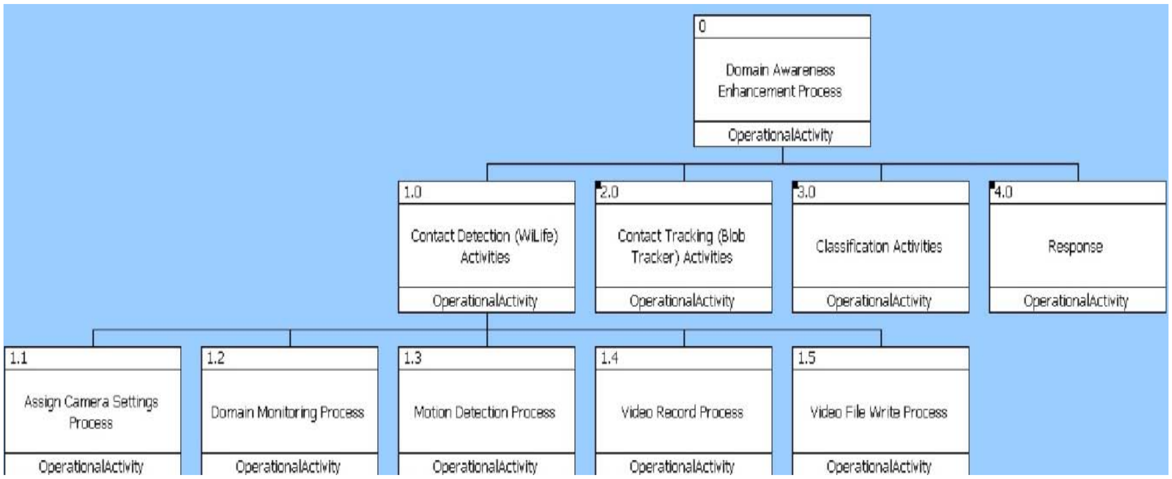


Figure 40. Detection Process – Level 1

Figure 40 depicts the decomposition of the first primary process, Contact Detection. All of the subprocesses for this process deal with the functioning of the camera system and video capturing functions that are required to accomplish contact detection.

What is not shown, but should be mentioned, is that some of the processes which fall under the Detection decomposition, could very well map to the Perform Overhead Functions function, and not necessarily to the Detect function, itself. This relationship illustrates that decomposing of processes and mapping those processes to functions are related, but not identical, activities. Processes are decomposed into the abstractions that aid in accounting for and allocating those processes to ensure their execution. However, to merely make the process decompositions mirror images of the functional decompositions would be a mistake of simplicity which could lead to extraneous processes and inefficient design.

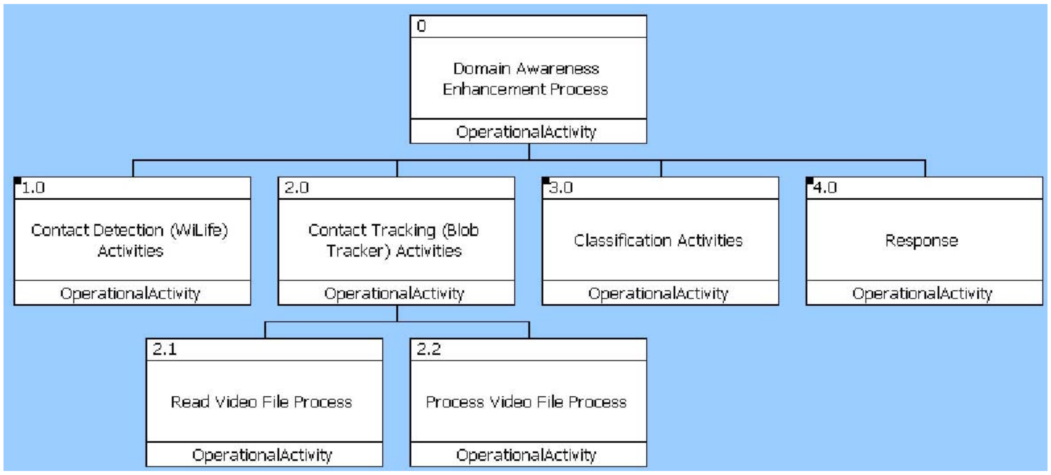


Figure 41. Tracking Process – Level 1

Figure 41 shows the Contact Tracking process, which is broken down into a Read Video File process and a Process Video File process. These processes are accomplished in the “Blob Tracker” software component of the WMSE POCS, which is fully described in Chapter V, and contain the activities required to conduct contact tracking for the system.

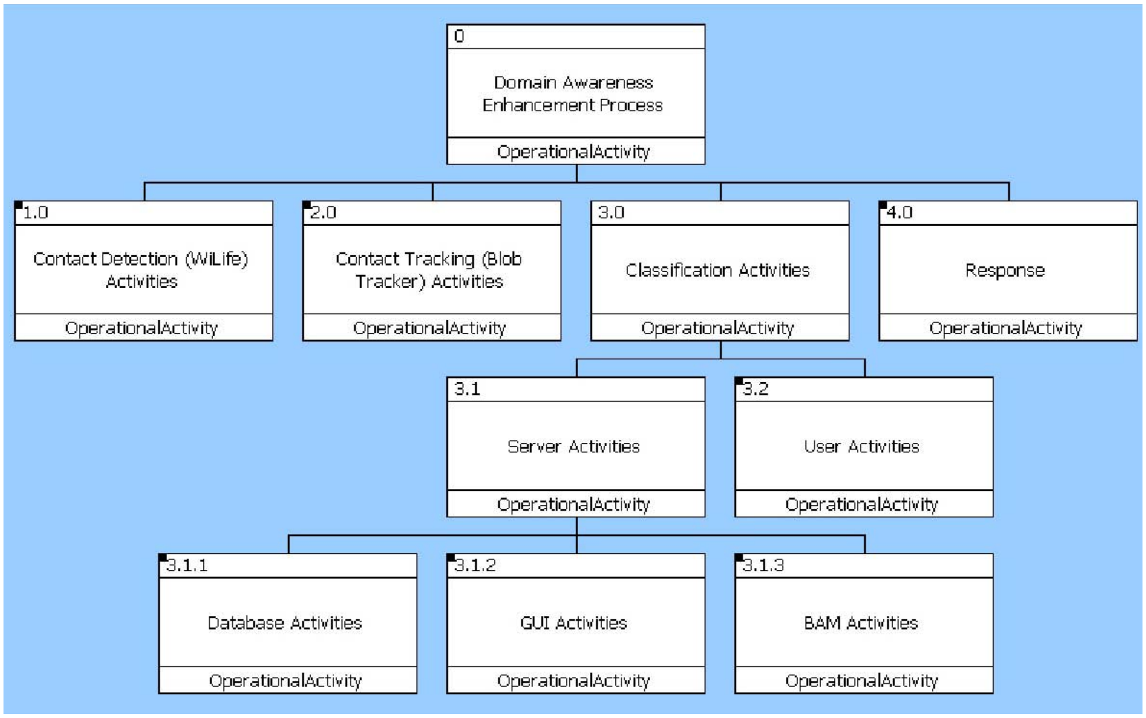


Figure 42. Classify Process – Level 2a

The Classification Activities process, shown in Figure 42, is broken down into two main sub processes, Server Activities and User Activities, determined as required to accomplish Classification. Figure 42 shows the further decomposition of one of these main sub processes, Server Activities, which is broken down into those activities the server needs to perform to support Classification, Database, Graphical User Interface (GUI), and Behavior Analysis Module (BAM) activities. All of these activities, and how they relate to functional achievement, are discussed in the system development presentation of Chapter V.

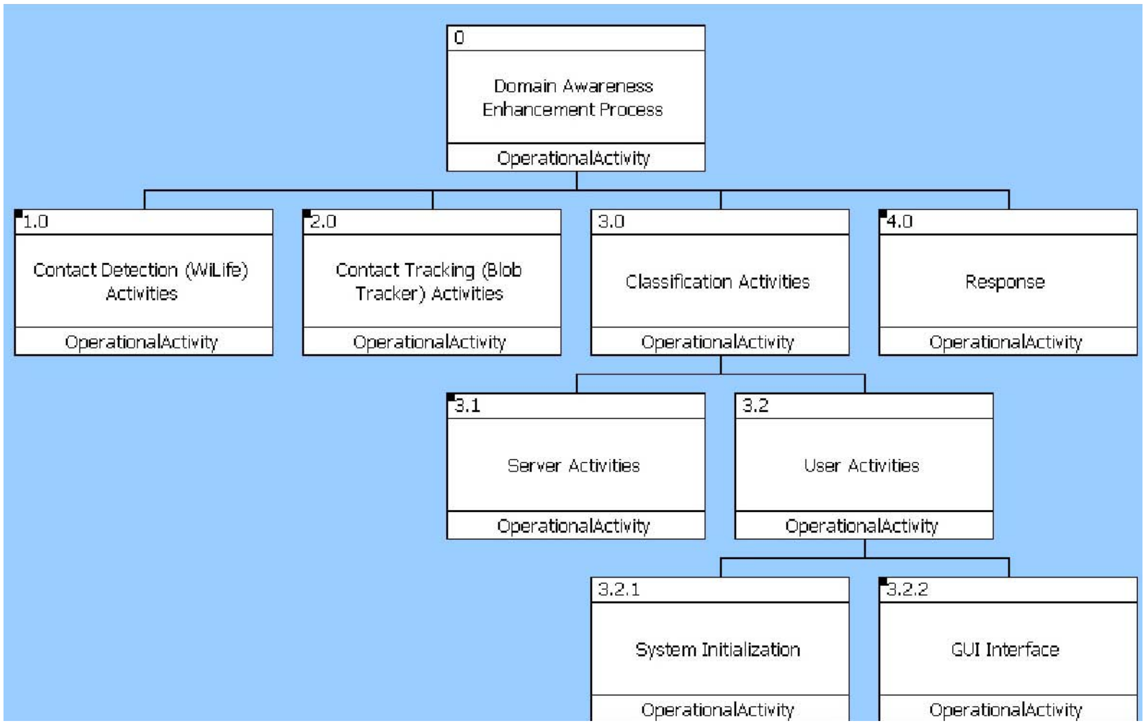


Figure 43. Classify Process – Level 2b

Figure 43 depicts the other main sub process of Classification, User Activities, which are System Initialization and GUI interface. These sub processes address those activities that the user must perform in order to accomplish various system functions.

Another point to mention is that ideally there will never be more than one process allocated to a function, as this would indicate an inefficiency of extraneous resources applied to execute a function. However, it is actually desirable to have more than one function allocated to a process, as this condition indicates greater efficiency with one activity accomplishing more than one function. While some multiple process to function relationships are unavoidable, they should be minimized to the maximum extent possible.

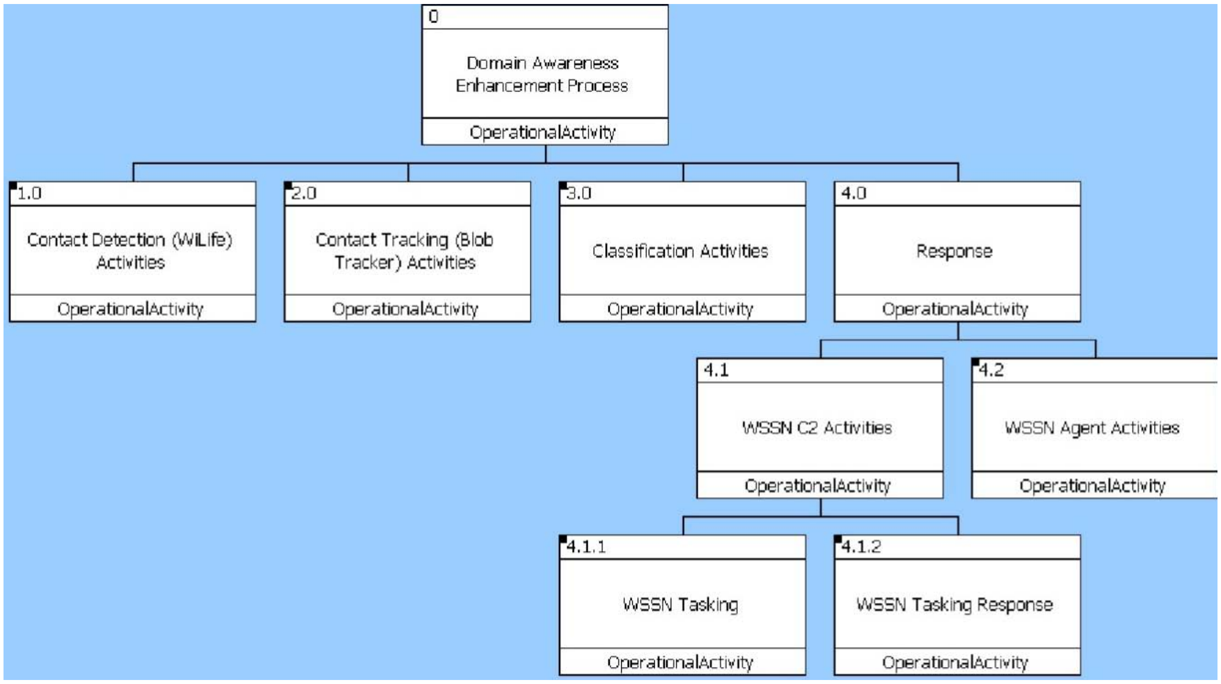


Figure 44. Response Process – Level 2a

The last main sub process, Response, is decomposed into two lower sub processes, WSSN C2 Activities, and WSSN Agent Activities. The decompositions of those lower sub processes are shown in Figures 44 and 45, with all of the depicted activities regarding the Wireless Smart Sensor Network (WSSN) described in Chapter V.

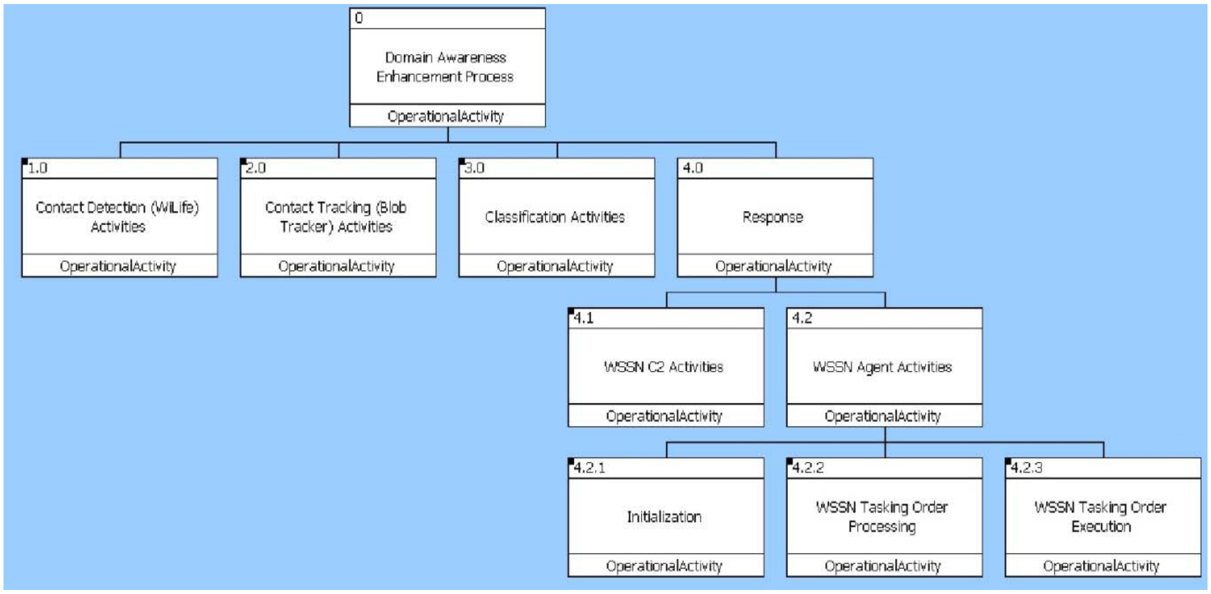


Figure 45. Response Process – Level 2b

3. Physical Integration Architecture

This subsection describes the processes and components that went into creating the physical system by translating the functional and process architectures into a physical architecture. This translation occurred through identifying the resources available, system boundaries, subsystem and element boundaries, and finally system constraints.

a. Integration Resources

In order to transform the system from a theoretical system to physical system, the physical architecture had to be defined and developed from the available technologies which were deemed the most feasible to accomplish the specific tasks, as well as those components which were selected under the process described in Section 4. The arrangement and layout of these physical elements is more thoroughly discussed in Chapter V. However, a short description of the resources used to accomplish the integration architecture is given here. The resources required to complete this project include the hardware and software that will be used to develop and integrate the desired functionality and further enhancements to the Watchman System. The resources for this architecture project are as follows:

- Network-Centric Systems Engineering Lab (BU-201L)
  - Watchman System
    - Watchman camera network
      - WiLife Cameras (6/node)
      - IEEE 802.3u compliant cabling and switches (see Option Matrix and Pugh Selection Matrix, Tables 6 and 7)
    - Watchman Server
      - MS Access Data Management System (DMS) Application
      - MATLAB BAM Classifier
      - MS SQL Server

        - Node Command and Control Processors
            - WiLife Command Center
            - Simulink Blob Tracker and associated MATLAB files
- Wireless Smart Sensor Network (WSSN)
    - Smart Sensor Network Agents
        - Lego Mindstorm NXT robots (2 - 4) with required sensors

b. System Boundaries

The Watchman Maritime Smart Environment is a network of cameras and desktop computers located in Bullard Hall. The system has six separate detection and processing nodes. Each node has six cameras connected to a dedicated desktop computer, referred to as the "Command Center," via Ethernet cables. The desktop computer for each node is connected to a single server.

This project focused on one of the six nodes, Node 6, and its interactions with the Watchman server. Node 6 receives input from its six cameras, three of which are all located in the "Smart Room," BU 201L, the Network Centric Systems Engineering (NCSE) Lab. The Node 6 Command Center and the Watchman Server are also physically located in the Smart Room. One unique feature of node 6 is that it contains two "Smart Agents." The Smart Agents are response robots built from Lego Mindstorm kits. They communicate with the Node 6 Command Center via Bluetooth, and accomplish the "Response" function (as a *React* subsystem), as noted in the functional architecture description.

c. Boundaries of Each Element

Smart Room: Entry to the NCSE Lab is controlled by key-card. The Watchman Server, the Node 6 Command Center, and the cameras are connected by a local private network but are also connected to the Internet. Penetration into the network by an outside party is unlikely but not impossible.

Cameras: The three Smart Room cameras of Node 6 are identical, and share a similar field of view, albeit from different perspectives: the center of the Smart Room. They are connected to the Node 6 Command Center via Ethernet cables and a network switch, which is also physically located in the Smart Room. All communication between the cameras and the Command Center is done via Internet Protocol (IP) through this switch, which physically connects Node 6 to the rest of the Watchman Network (five other Nodes, each with their own sets of cameras). Although the Node 6 Cameras are connected to the rest of the network, the cameras do not communicate with any of these other nodes or other external networks under the scope of this project.

Node 6 Command Center: The Command Center is connected to the Watchman Server and Network via a server switch as well. The Node 6 Command Center does not communicate with any of the other Nodes for the purposes of this project (other than administrative file sharing). Though the Command Center can access the Internet, Internet access is not used for any of the Watchman processes. The Command Center Communicates with the Smart Agents via Bluetooth communication. Bluetooth is not used for any other purpose.

Watchman Server: During normal use, the Watchman Server communicates with the six nodes via a server switch. For this project, the only focus regarding communication is between the Watchman Server and the Node 6 Command Center. Though the Watchman Server can access the Internet via a router, Internet access is not used for any of the Watchman processes.

Smart Agents: The Smart Agents are equipped with color sensors, ultrasound range sensors, and a camera (on only one agent). They communicate with each other and the Command Center via Bluetooth. They do not communicate with any other system or entity.

Bluetooth: This technology is widely used. The Command Center and the Smart Agents do not use encryption in data transfer. Another interested party on or off campus could intercept Bluetooth data; since radiation from Bluetooth sources can be detected many miles away (and may even be picked up by a satellite in orbit). . Bluetooth is also vulnerable to interference, either by cross talk from other sources or deliberate jamming.

More details of the physical architecture, layout, design, build, and test demonstrations of the WMSE POCS, including the WSSN are found in Chapter V.

d. System Constraints

Cameras and Command Center: The Cameras use a proprietary program called “WiLife” which handles the video feed by saving it in video frames of 10-15 seconds in length. So far, a solution, which provides a direct real-time streaming video feed for analysis, has not been found.

Watchman Server: The Watchman Server uses the Mac OSX operating system. The Watchman *Identify*, *Predict* and *React* software is run through Microsoft SQL, Microsoft Access, and MATLAB and must be run through a Windows virtual machine. This uses a great amount of processing power when analyzing the video data and could prove problematic when attempting to process a large amount of video. The server is also currently unable to communicate directly with the Smart Agents via Bluetooth and must rely on one of the Command Centers to order to communicate with the WSSN agents. The server can then communicate with the Command Center via IP.

Smart Agents: The two Smart Agents can communicate with the Command Center via Bluetooth only. Over Bluetooth, the transfer rate of data is extremely limited. This limits transmission of video collected by one of the agents to low resolution and low frame rate.

This chapter intended to describe and present the functional, process, and operational architectures that were produced as part of the functional analysis and allocation steps of the SE process to show how the resultant architectures were ultimately applied in the design, building, coding, integration, and testing of the WMSE POCS.

The next chapter presents, and describes in detail, this system, as well as conducts an analysis and makes recommendations for operational implementation of automated behavior analysis capability represented in the system.

V. WMSE POCS AND OPERATIONAL IMPLEMENTATION

Chapter V presents and describes the elements and development of the proof of concept (small-scale laboratory) system (POCS) that was dubbed the “Watchman Maritime Smart Environment (WMSE) in the context of this research. This description includes the hardware and software design, layout (topology), and build, as well as improvements and upgrades made to the system after the initial development. In the final section, the chapter discusses the lessons learned from the system design, development, and integration processes, analyze alternatives for potential operational implementation, and make recommendations for the preferred approach. In addition, the section discusses what technology still needs to be refined for potential application and implementation of the system in an operational environment.

The WMSE POCS was born from an elective course in the Network-Centric Systems Engineering Track at NPS, Artificial Intelligence (AI) Systems I, and was improved and expanded over subsequent, follow-on courses (AI Systems II & III), as well as through directed study course blocks. The AI Systems class series was designed to apply the systems engineering process, through artificial intelligence technologies, to design, implement and demonstrate a “smart environment” through behavior analysis. The courses were formatted as a “skunk-works” type project, and carried the project from concept to prototype demonstration, on through Engineering Change Orders (ECO) and system upgrades and modifications. The system was divided into three subsystems – Applications Engineering, Network Engineering, and Software Engineering – with student teams responsible for each subsystem development. The SE process for the system development was fully documented along the way: CONOPS, Request for Proposal, Proposal, Preliminary Design Review, Critical Design Review, System Documentation, Demonstration Proposal. The class had several team review, management and interactive meetings while designing and implementing this interdisciplinary system.

My role was the project manager (PM) and team lead for the Applications Engineering Subsystem team. In this role, I coordinated all of the activities mentioned previously, leading the team through all aspects of the SE process, which was discussed in Chapter III, as well as ensured proper documentation of all of these activities. The role of PM for the Applications team was unique from the other team leads, in that the Applications group, as will be shown, was the focal point and coordinating hub for the realization of the system's main goals. It was imperative for all subsystem teams to coordinate with one another to ensure accurate data flow and correct interfaces between subsystems; however, the Applications team was responsible for bringing all of the pieces of the system together to make certain that the system was fully integrated and that exact information flowed to the right parts of the system in the right way, at the right time.

Due to the fact that the Applications subsystem was where the behavior analysis and processing took place, I also became intimately familiar with the ABA algorithms (known as the Behavior Analysis Module, or BAM), and became the chief software engineer for implementing, upgrading, and expanding the BAM for operation in the WMSE POCS. It is with this experience, coupled to my operational Naval Aviation and Carrier Battle Group experience, and my professional acquisition experience that have all contributed to the insight in applying this research to the challenges of MDA.

The entire system is presented and explained here to provide a notional "operational network" context for the basis of the research. My specific research effort in this context involved integrating the BAM within the Watchman network system. Although I was certainly involved in all aspects of system design and development, my area of concentration for this thesis was the portion of the system that dealt with the design, integration and performance of a behavior analysis capability in a networked environment. Additional contributions in this regard included the design, development, and integration of a Wireless Smart Sensor Network (WSSN) to demonstrate how behavior analysis output could be integrated with a networked complex reaction subsystem; and design and development of a spatial resolution enhancement capability, and automated cost calculation algorithms for more discrete behavior analysis.

A. WMSE POCS

This section includes an overview of the initial Watchman system, a description of each subsystem and its functions, and finally a description of the system upgrades, enhancements, and additions that were accomplished.

1. Watchman System Overview

This Watchman System (referred to as Watchman) is an automated “smart” system, which continuously monitors designated areas in order to identify pre-determined individuals or for classifying observed behaviors and alerting operators of observed behaviors. This automated system allows users to react, in a timely manner, to “abnormal” behaviors in order to minimize the potential simulated threats that certain behaviors could pose. Many different environments, such as insurgent urban areas, military ships and installations, temporal “high-value target” areas (e.g., gatherings of international leaders, political events, high-profile sporting or entertainment events, events with dignitaries and/or potential targets of terrorist elements, etc), schools, government buildings, and others, all require this type of constant behavior surveillance and monitoring.

The initial system was able to identify, classify, and track certain individuals (class members) for accountability purposes. Those individuals were identified by the system and their presence in the test environment was then updated into an accountability (mustering) system/database. Once the accountability baseline had been established for the indentified individuals, that control group could be monitored in the test environment in order to analyze behaviors to collect data on differences between normal and abnormal behaviors.

The system was able to classify behaviors as Normal, Abnormal and Unknown. Upon the system gaining enough data to discern which behaviors can be classified, the operators could then add additional behaviors into the Watchman behavior database. This gives Watchman the ability to adapt to its environment, which would offer system stakeholders additional mission adaptability.

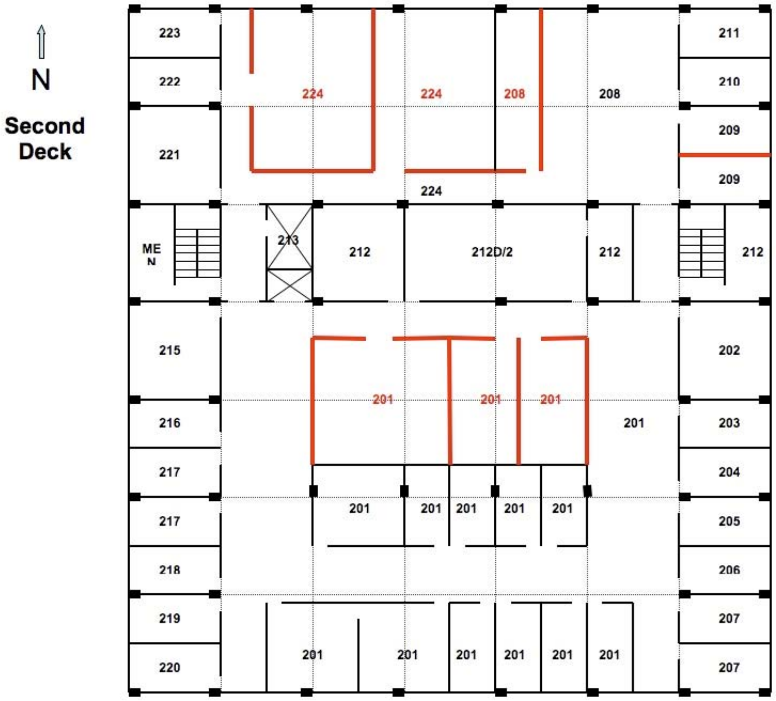


Figure 46. Bullard Hall, Second Floor

The system has the ability to monitor the control group individuals and send alerts, via the server subsystem, to operators to react to those behaviors. The system retains data on the types of behaviors which were classified as abnormal, and when alerts were sent.

Watchman was set up on the Naval Postgraduate School campus on the second floor of Bullard Hall (see Figure 46).

The original Watchman was comprised of three subsystems: Watchman Application Engineering Subsystem (WAES); Watchman Network Engineering Subsystem (WNES); and Watchman Software Engineering Subsystem (WSES). The architecture can be seen in Figure 47.

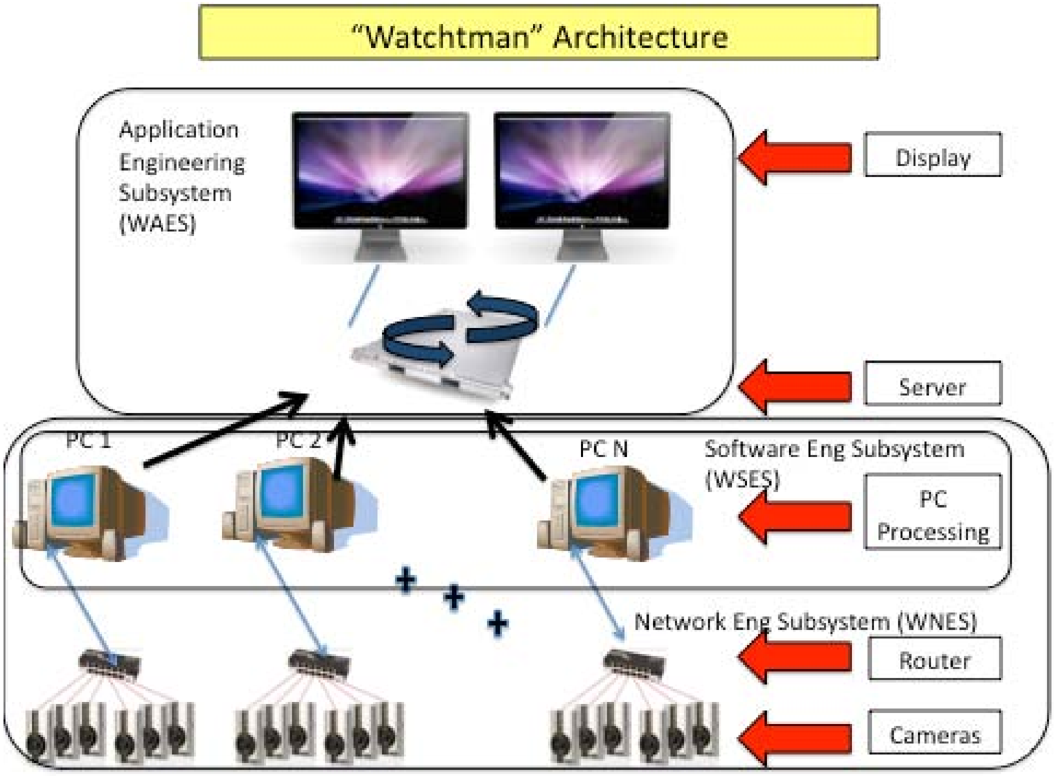

"Watchman" Architecture
Application Engineering Subsystem (WAES)
Display
Server
PC 1
PC 2
PC N
Software Eng Subsystem (WSES)
PC Processing
Network Eng Subsystem (WNES)
Router
Cameras

Figure 47. Watchman Architecture

The WAES contains the command-and-control center that has two large (30" cinema) displays that allows the operators to control and monitor the system. There is an Apple "X-Server" that is interconnected for reliability and that processes the data files and video feed pushed by the WNS.

The WNES provides an Internet Protocol (IP) network engineering physical infrastructure to capture video data and relay this data to the command center computers for analysis. This infrastructure includes a link between the command and control center computers and the master server. The infrastructure consists of IEEE standardized Ethernet networking tools—to include cabling, switches, and network interface cards—as well as surveillance cameras and data analysis software.

The WSES requires raw video data input from the surveillance cameras. Data transfer is provided by the network subsystem. The detection and identification software at each PC node for processing receives this data. The WSES was designed to initially process detection and identification of one (1) person at a time to provide facial recognition for identification and mustering and for behavior analysis.

The detection and identification software processes the input video data to extract meaningful identification and tracking information. This data is aggregated into an Structured Query Language (SQL) database, whose data is pushed from each PC node to the master server database.

The software subsystem provides archived video data to the server via the network subsystem. The camera video data is stored by the WSES to allow on-demand access by the WAES via the network subsystem. Additionally, continuous real-time streaming video data is available to the server via the network subsystem.

2. Application Engineering Subsystem

This subsection describes the Application Engineering Subsystem within the WMSE POCS.

a. Application Engineering Subsystem Overview

The Watchman Application Engineering Subsystem (WAES) collects, collates, and processes target movement and identification data. The WAES displays alerts, identifications, behaviors, and movement patterns to the Watchman operators using a graphical user interface (GUI) on the command and control displays that are located in Bullard Hall room 201L.

The WAES receives all data feed from the Watchman Software Engineering Subsystem, through the Watchman Network Engineering Subsystem infrastructure. Each WSES PC node continually pushes data to the master database detection table located on the WAES. The detection table was originally comprised of data entries that contain the following information: Data Type (Position or Identification); Identification (Name); Node and Camera; Figure of Merit (FOM); Date and Time. Database updates are pushed to the WAES at variable times (as determined and set by the operator). Database information collection can also be configured to initiate whenever WSES detects an individual in the surveillance area.

The database information is used in two ways. First, the ID Module can query the database detection table in order to create muster alerts and a muster report. Second, the Position Integration Module (PIM) can query the detection table periodically at pre-user-determined time intervals in order to create a time/position vector for use in the Behavior Analysis Module (BAM). During this vector creation query process, the query will analyze information in the database using the Position De-confliction Module (PDM). If an individual is detected and reported simultaneously by two different Nodes, both nodes will send a database input to the WAES. If these duplicate inputs have the same time data field, the WAES shall select which Node has the higher FOM, and use that input as the data for that particular time period in the creation of the time/position vector for behavior analysis.

b. Behavior Analysis

From the vector creation query, the PIM will process the database query and will order the Node and Camera data, by time sequence, into a data array that will hold the positions of a contact (identified or un-identified), integrating data over time in order to give a slower and more accurate position account of the person being tracked. This data set will be written to three separate text files, stored on the server. The first file will contain a string of formatted time and date data, the second will contain a string of Node data, and the third will contain camera identification data. These three files will be formatted such that when they are processed, the time and date data in the first file will correlate directly to the Node and camera that captured the video data at that particular time. The BAM will continually query the storage location directory for the data files. When the PIM writes new files to the directory after a database query, the BAM will pull the most recently updated files for processing. See Figure 48 for graphical view of the data flow.

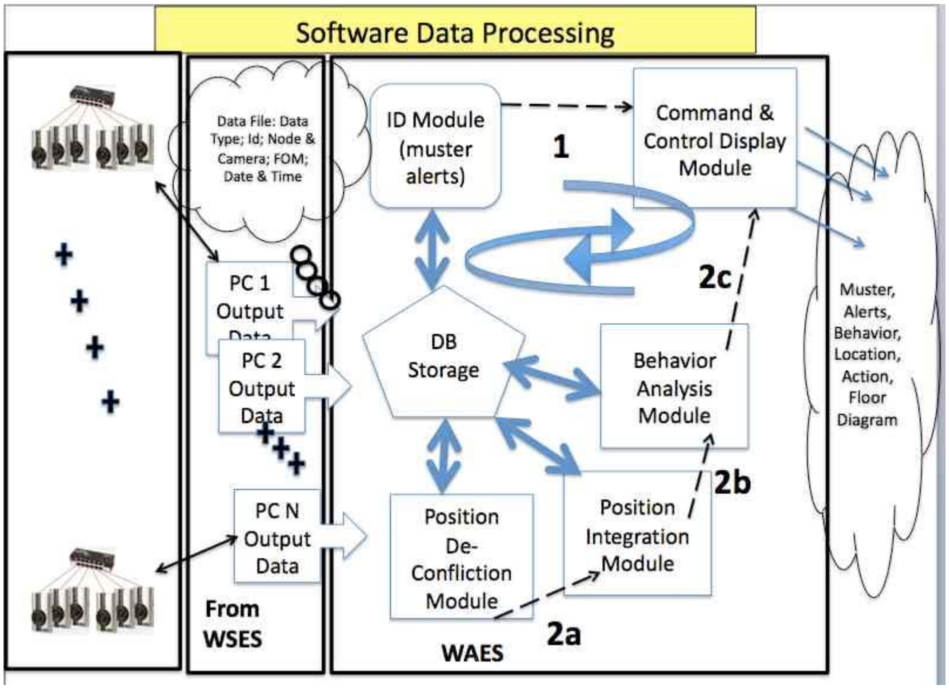

Software Data Processing
Data File: Data Type; Id; Node & Camera; FOM; Date & Time
ID Module (muster alerts)
Command & Control Display Module
1
PC 1 Output Data
PC 2 Output Data
DB Storage
2c
Behavior Analysis Module
Muster, Alerts, Behavior, Location, Action, Floor Diagram
2b
PC N Output Data
Position De-Confliction Module
Position Integration Module
From WSES
WAES
2a

Figure 48. Software Data Processing

The BAM (Figure 49) will process the data files from the PIM, and classify the observed behaviors according to the following sequence: The BAM will compare the observed behavior to the stored behavior model using a sequential syntactical classifier algorithm, such as was discussed in Chapter III. This algorithm is based upon alphabetical syntax rules. An alphabet of symbols is pre-defined corresponding to node/camera combinations. Each behavior is defined by augmented syntax rules (e.g., sequences of similar structures) for the symbols. When a sequence to be classified is read into the program, it is parsed according to each predefined behavior, which edits the sequence to fit those behaviors. This “fitting” process assigns a cost (or distance) according to the amount, which the parsed string was changed in order to match each behavior model. The algorithm then outputs this distance metric to the Tolerance Comparator (TC). The behavior separation distances will then be compared by the TC to ensure that they are smaller than a maximum “threshold”/baseline distance. If the behavior distance is less than the threshold, then the behavior will be classified as either normal known behavior, giving the behavior label, *I*, that corresponds to the min(d_i), or a known abnormal behavior (which would be one of the predefined Behavior , where *I* =

1,2,…,*N*). If the behavior distance is greater than the tolerance, then the behavior will be classified as unknown (where unknown could be an "unknown normal behavior" or an "unknown abnormal behavior").

The data output from the BAM will be in the form of a text file that will be the result of the analysis of the data input from the PIM files. This output shall be one of four possibilities: NORMAL, ABNORMAL, UNKNOWN, or ERROR. The first three possibilities listed will merely be the classification of the PIM data by the BAM. The final possibility, ERROR, will be the output if the BAM cannot resolve the input data to any of the other three possibilities, due to failure or limitation of the algorithm or erroneous input data from the PIM. This output will be stored in the database with a date-time tag in order to have a reference to regenerate the PIM query for analysis, if required.

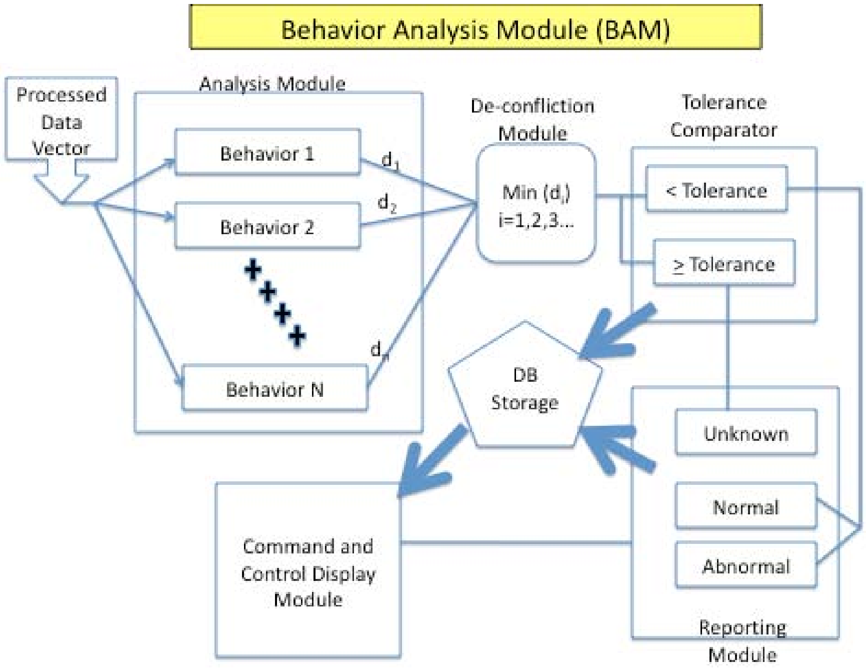


Figure 49. Behavior Analysis Module

The three general behavior classifications: Normal, Abnormal and Unknown are defined below.

- Normal Behavior – behavior that the customer/project team deems to be acceptable or non-threatening and does not require alerting, further tracking, or reporting.
- Abnormal Behavior – behavior that the customer/project team deems to be non-acceptable or potentially threatening and requires alerting, immediate further tracking and reporting.
- Unknown Behavior – behavior that does not meet the thresholds of predefined normal or abnormal, and thus will be reported to the operators for analysis and further tracking and will be used in the future for adaptive learning of the defined behaviors, behavior parameters, and TC parameters.

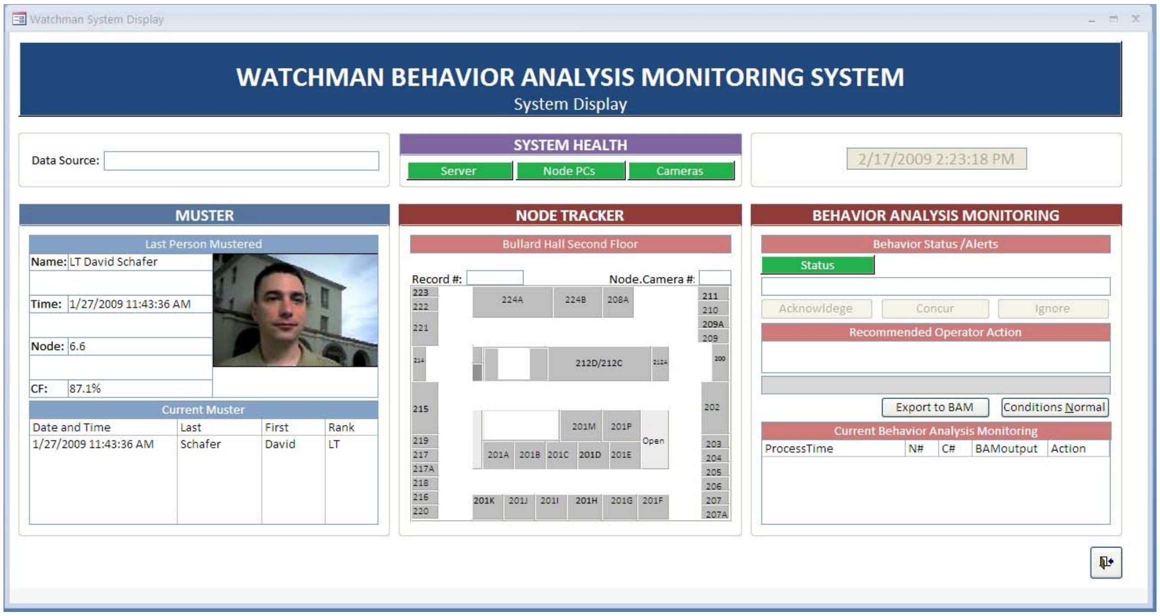
c. Identification and Mustering

If the data from the WNS has the Identification (ID) data field populated (Face Recognition has made a match), then it will be passed onto the ID Module in Figure 3. The ID Module will check the data file ID and compare it to the Daily Muster List (DML). The DML will be active for a 24-hour period starting at 00:00 local time. The DML will have all the identified contacts and the date/time stamp of their first identification during the 24-hour period. The ID Module will check the ID of data file and will enter the date/time stamp if there is not one for the current 24-hour mustering period, or discard the data file if a date/time stamp already exists in the DML. The DML will update the DB storage area after each update of the DML.

d. Command and Control

The Command and Control Display Module (C^{2}DM) consists of two (30" cinema display) monitors that use a GUI to display alerts of any abnormal or unknown behavior. The C^{2}DM displays the global view of distributed IP smart camera network, for situational awareness of the area of interest (Bullard 2nd floor). The GUI allows operators to access and view the streaming video feed from any of the video cameras in each node. The C^{2}DM design encompasses two main interfaces, which correspond to a user-defined preference to reside on either of the two displays. The two interfaces are the System Display Interface (SDI) and the Video Display Interface (VDI). The CDI is a MS Access based form, which will display several system parameters, status, and alerts, as well as to provide a user interface to assimilate, process, and react to the data output of the Watchman system. The VDI provides the capability to view video data from each Node via the Wi-Life web interface.

(1) System Display Interface (SDI). The SDI has several graphical and user-interactive interfaces shown in Figure 50.



Watchman System Display

WATCHMAN BEHAVIOR ANALYSIS MONITORING SYSTEM
System Display

Data Source:

SYSTEM HEALTH

2/17/2009 2:23:18 PM

MUSTER

Last Person Mustered

Name: LT David Schafer

Time: 1/27/2009 11:43:36 AM

Node: 6.6

CF: 87.1%

Current Muster

Date and Time	Last	First	Rank
1/27/2009 11:43:36 AM	Schafer	David	LT

NODE TRACKER

Bullard Hall Second Floor

Record #:

Node.Camera #:

BEHAVIOR ANALYSIS MONITORING

Behavior Status /Alerts

Recommended Operator Action

Current Behavior Analysis Monitoring

ProcessTime	N#	C#	BAMoutput	Action

Figure 50. System Display Interface

(a) SDI Tracking Module

The SDI Tracking Module, which consists of an interactive map display of the second deck of Bullard Hall, takes data from the PIM, and resolves it to a regional position on the map corresponding to the camera field of view (FOV) for the node in which the PIM reports the target as being detected. The position on the map will be indicated by a time-phased color scheme; black depicting the current position of the target, red being the first time-late position, orange depicting the second time-late position, and yellow depicting the third time-late position.

(b) SDI Alert Module

The SDI Alert Module takes the output from the BAM described above, being triggered by the input of the BAM output data into the PIM database table. The SDI uses this data for the alert function. When data is written to the PIM database table, the SDI analyzes the input. If the input is "NORMAL," the system STATUS indicator will remain green, and no other alerts will be shown. If the input is "UNKNOWN," the STATUS indicator will turn YELLOW, the ALERT indicator will appear and be YELLOW, and the ACKNOWLEDGE (ACK) button will become active. The alert text field under the ALERT indicator will read "UNKNOWN BEHAVIOR DETECTED," and the Operator Action text field above the ACK button will read, "INVESTIGATE TARGET EXHIBITING UNKNOWN BEHAVIOR IN NODE (X)," "X" being the node in which the target is currently located by the PIM. These indications will remain until the operator initiates the ACK command. Upon initiation of this command, the ALERT will disappear to indicate that the operator has acknowledged the ALERT, but the STATUS will remain YELLOW until the system is cleared "CONDITIONS NORMAL" by the operator. In addition, two additional, follow-on functions shall become active; CONCUR, and IGNORE. If the IGNORE function is selected, the STATUS will return to GREEN and the system will revert back to "CONDITIONS NORMAL." If the CONCUR function is selected, the STATUS will remain its current color until the system is cleared "CONDITIONS NORMAL" by the operator. If the input from the BAM is "ABNORMAL" the system will behave identically to the "UNKNOWN" condition, with the exceptions that the ALERT and STATUS indicators will be RED, and the Alert text field will read, "ABNORMAL BEHAVIOR DETECTED" and the Operator Action text field will read, "INTERCEPT TARGET EXHIBITING ABNORMAL BEHAVIOR IN NODE (X)." At any time, the operator can activate the CONDITIONS NORMAL function to return the system to the normal operating mode. In future increments, the CONCUR and IGNORE functions could incorporate the additional capability of providing input to the BAM as to the validity and accuracy of the reported behavior in order for the BAM to "learn" and continually refine its analysis algorithm.

(c) SDI Mustering Module

The SDI displays the information output from the ID Module when a target is identified and their data is input to the database detection table. The SDI will display the stored file picture of the individual who has just been added, as well as a "muster data tag" (MDT) which consists of their name, rank, date and time of data capture, Node and Camera, as well as a confidence factor at the moment data capture occurred. Simultaneously, the MDT will be added to the DML, which will be displayed under the incoming MDT and current mustered picture. As additional individuals are identified and mustered, the most current data will replace the picture and MDT, as well as to be added cumulatively to the DML.

e. Application Engineering Subsystem Detailed Engineering Specifications

This subsection itemizes the detailed engineering specifications for the WAES, listed as enumerated system requirements.

Requirement 1.0:

1.1: The system shall have the ability to receive near real-time video and/or feed from any camera node 95% of the time requested.

1.2: The system shall successfully receive data from camera nodes (vector information and time data) 95% of the time requested.

Requirement 2.0:

2.1: The system shall meet a threshold requirement to be able to successfully identify and match five recognized behaviors (3 "normal" behaviors and 2 "abnormal" behaviors) to pre-known behaviors 60% of the time it is requested. A future objective of the system is to be able to identify and match ten behaviors (6 "normal" and 4 "abnormal") to pre-known behaviors 80% of the time it is requested.

2.2: The system shall successfully resolve data conflicts between camera nodes 85% of time requested in order to maintain track integrity.

2.3: The system shall successfully resolve and correct of face recognition data 85% of time requested in order to maintain mustering integrity.

2.4: The system shall be able to successfully report the presence of control group personnel, via facial recognition data from the PC nodes and error correction, to a personnel accountability system 80% of the time requested.

2.5: The system Command Control Center shall maintain and display near real-time mustering of the area of interest (Bullard 2nd floor) as desired by the system operator, 100% availability.

2.6: The system shall be able to successfully classify target behavior into three categories ("normal," "abnormal" and "unknown") 80% of time requested. This data will be used identify key metrics and further develop the systems behavioral analysis.

2.7: The system will have an open architecture to allow for future growth of the system's ability to "learn" and auto-update the behavior classification matrix based on collected data from PC tracking and reporting and operator inputs. Therefore, with this system open architecture, the system has the ability to scale to additional behaviors and in the future adaptively learn and update these behaviors, the behavior parameters, and the TC parameters.

Requirement 3.0:

3.1: The system shall have the ability to command the PC nodes to send raw video data (near real-time data or archived data) 95% of the time necessary. This data shall be able to be displayed to the operator at the Command Control Center.

Requirement 4.0:

4.1: The system shall have a 90% rate of successfully transmitting an alert of reportable behaviors to an operator at the Command Control Center. The future objective of the system is to be able to successfully transmit an alert of reportable behaviors with 100% accuracy.

4.2: The system shall be able to continue normal operations (track and reports further reportable behaviors) while prior alerts are being addressed 90% of the time.

Requirement 5.0:

5.1: The system's Command Control Center design shall be user-friendly and require minimal operator training.

5.2: The system's User Interface will be user-friendly, allow users to locate a person's general position (within the range of one node) on Bullard Hall's 2nd floor, and be capable of displaying near real-time video.

5.3: System data shall be classified and stored on a designated server indefinitely. Operators shall be able to view previously recorded video data, as well as near real-time data. Server back-up capability, or PC reach-back capability, will allow the WAES to get database files from each WSES node. This will protect against lost information.

f. System Demonstration / Subsystem Acceptance Test Procedure (ATP)

This ATP was developed for the purpose of demonstrating the system to complete the validation phase of the SE process. This particular ATP is a system demonstration, which validates the functionality of the WAES. Due to the integrated nature of the system, this ATP also validates the functionality of the WNES, and WSES, and therefore is a validation of the entire system.

System Test Scenarios – Data output via GUI is collected in an output data file, which provides metric reports on the numbers and types of identifications, classifications, alerts, and system errors. Scripted sets should be run in the following order for the purpose of data analysis and comparison to expected results:

FACIAL RECOGNITION:

a. Subject will walk within identification range of sensor node (X), and continue through node at a normal walking pace until out of the node. During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute.

b. Upload stored image of test observer. Have observer look into operator station camera to be collected, identified, and mustered in the system.

* Successful test criteria:

1) Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification that individual has been mustered.

2) Identification Alert shows the correct name, picture, date and time information, matched to the subject, date and time of test run

3) Output data file shows subject in the output data file muster list

4) Output data file shows correct name, date and time information, matched to the subject, date and time of test run.

5) Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

NORMAL BEHAVIOR – 1 NODE:

Subject walks through Node (Y) along a planned route, corresponding to a NORMAL behavior pattern, at a normal walking pace. During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (Y).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of NORMAL BEHAVIOR.

b. Output data file shows subject identified (if Node (Y) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen “map” view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

ABNORMAL BEHAVIOR –1 NODE:

Subject walks through Node (Y) along a planned route, corresponding to an ABNORMAL behavior pattern, at the proscribed walking pace. During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (Y).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of ABNORMAL BEHAVIOR.

b. Output data file shows identified subject (if Node (Y) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen “map” view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

UNKNOWN BEHAVIOR – 1 NODE:

Subject walks through Node (Y) along a planned route, corresponding to an UNKNOWN behavior pattern, at the proscribed walking pace. During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (Y).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of UNKNOWN BEHAVIOR.

b. Output data file shows subject identified (if Node (Y) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen “map” view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

NORMAL BEHAVIOR – 2 NODES:

Subject walks through Nodes (Y and Z) along a planned route, corresponding to a NORMAL behavior pattern, at a normal walking pace. During subject conduct of test run, operator will send out command for video feed to control station.

Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (Y or Z).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of NORMAL BEHAVIOR.

b. Output data file shows subject identified (if Node (Y or Z) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen “map” view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

ABNORMAL BEHAVIOR – 2 NODES:

Subject walks through Nodes (Y and Z) along a planned route, corresponding to an ABNORMAL behavior pattern, at a normal walking pace. During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (Y or Z).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of ABNORMAL BEHAVIOR.

b. Output data file shows subject identified (if Node (Y or Z) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen "map" view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

UNKNOWN BEHAVIOR – 2 NODES:

Subject walks through Nodes (Y and Z) along a planned route, corresponding to a UNKNOWN behavior pattern, at a normal walking pace. During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (Y or Z).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of UNKNOWN BEHAVIOR.

b. Output data file shows subject identified (if Node (Y or Z) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen "map" view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

NORMAL BEHAVIOR – MULTIPLE NODES:

Subject walks through Nodes (Y through Z via C and D) along a planned route, corresponding to a NORMAL behavior pattern, at a normal walking pace.

During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (C, D, Y or Z).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of NORMAL BEHAVIOR.

b. Output data file shows subject identified (if Node (C, D, Y or Z) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen “map” view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

ABNORMAL BEHAVIOR – MULTIPLE NODES:

Subject walks through Nodes (Y through Z via C and D) along a planned route, corresponding to an ABNORMAL behavior pattern, at a normal walking pace. During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (C, D, Y or Z).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of ABNORMAL BEHAVIOR.

b. Output data file shows subject identified (if Node (C, D, Y or Z) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen “map” view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

UNKNOWN BEHAVIOR – MULTIPLE NODES:

Subject walks through Nodes (Y through Z via C and D) along a planned route, corresponding to an UNKNOWN behavior pattern, at a normal walking pace. During subject conduct of test run, operator will send out command for video feed to control station. Operator shall monitor video at their discretion for no less than 1 minute. Test shall end upon subject exiting personnel recognition range of Node (C, D, Y or Z).

* Successful test criteria:

a. Identification alert passed to operator screen with name, date/time stamp, video capture image, and stored file picture, with text notification of UNKNOWNBEHAVIOR.

b. Output data file shows subject identified (if Node (C, D, Y or Z) has facial recognition capability) and contains behavior classification data.

c. Output data file shows correct name, date and time, and behavior classification information, matched to the subject, date and time of test run.

d. Data output to operator screen “map” view, which shows correct position information of subject throughout conduct of test.

e. Video is passed to operator display in an uninterrupted data stream, which provides a clear, discernable image of the subject and/or test node area.

End of technical demonstration description

Upon execution of the system demonstration test procedure for all cases, the system passes all test criteria, correctly identifying individuals for mustering 90% of the time (9/10 test subjects correctly identified), and correctly identifying observed behaviors in each test run, for 100% behavior classification. These results exceeded system requirement metric thresholds.

3. Network Engineering Subsystem

This subsection describes the Network Engineering Subsystem within the WMSE POCS.

a. Network Engineering Subsystem Overview

The Watchman Network Engineering Subsystem (WNES) provides a Commercial Off The Shelf (COTS) IP network engineering physical infrastructure to capture video data and relays this data to the command center computers for analysis. This infrastructure includes local area network links between the command center computers and the master server of the WAES, described in the previous section. The infrastructure consists of IEEE standardized Ethernet networking tools—to include cabling, switches, and network interface cards—as well as WiLife cameras and software. Figure 51 depicts the networked components that make up the WNES.

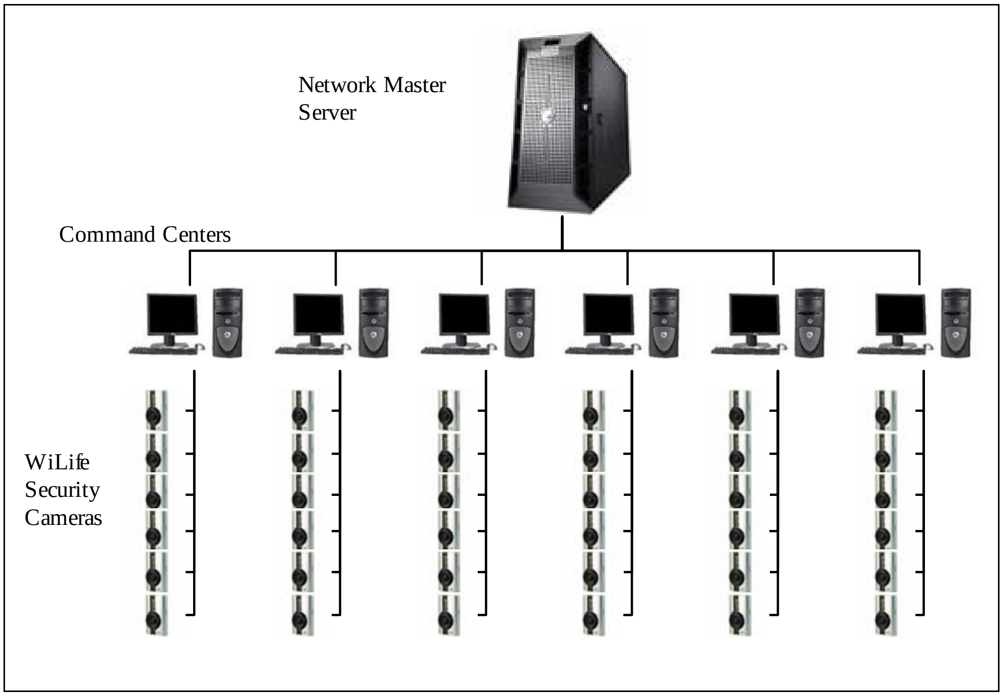


Figure 51. Network subsystem component overview

b. *Physical Network Topology*

The Watchman system uses a switched Ethernet network with the capability of being connected to another network or the Internet. All cables are Cat 5e compliant, and the network will operate at 100Mbps utilizing IEEE 802.3u compliant hardware, as per the component selection process outlined in Chapter IV (Tables 6-7). Figure 52 shows a diagram of the network topology. Each of the six cameras in one group connects to the WiLife "power injector," a 12-port switch that provides Power-over-Ethernet to the cameras per the IEEE 802.3af standard. The remaining six ports in the "power injector" switch connected via short patch cables to a D-Link 8-port switch. One of the remaining ports in the D-Link switch connects to the "command center" PC Central Processing Unit (CPU) for that group of cameras, and the other port is connected to a Linksys 8-port switch. This switch is connected to the switch for each camera group and to the WAES Mac (Apple) server. This switch is also connected to the NPS network via the Linksys router to provide access to the Internet. Both the D-Link and Linksys switches are IEEE 802.3u compliant.

The connectivity components within this topology are an example of how the physical system components can be traced back to the user need via the SE process defined in this thesis. Beginning with the user need, “Push time-critical alerts to decision makers,” MDA-003C-007T, from the JIC, the operational activity, “Assess the Operational Situation” (COM 1.1.1.4.3) was derived and then mapped to the identified Navy task, “Provide I&W of Threat” (NTA 2.4.5.3). This task was then mapped to the function, “Perform I&W” (3.0) and its constituent subfunction, “Alert Generation” (3.2) from the functional architecture, both derived from the Network Engineering Subsystem requirement, “Provide stable high-speed (at least 100Mbps) data connectivity for timely generation of alerts.” This requirement led to the criteria for the Morphological Matrix of Options shown in Table 6, found in Chapter IV. From this analysis of options, the network components of 802.3u IP and Integrated Network Card were selected and implemented in the Network Topology.

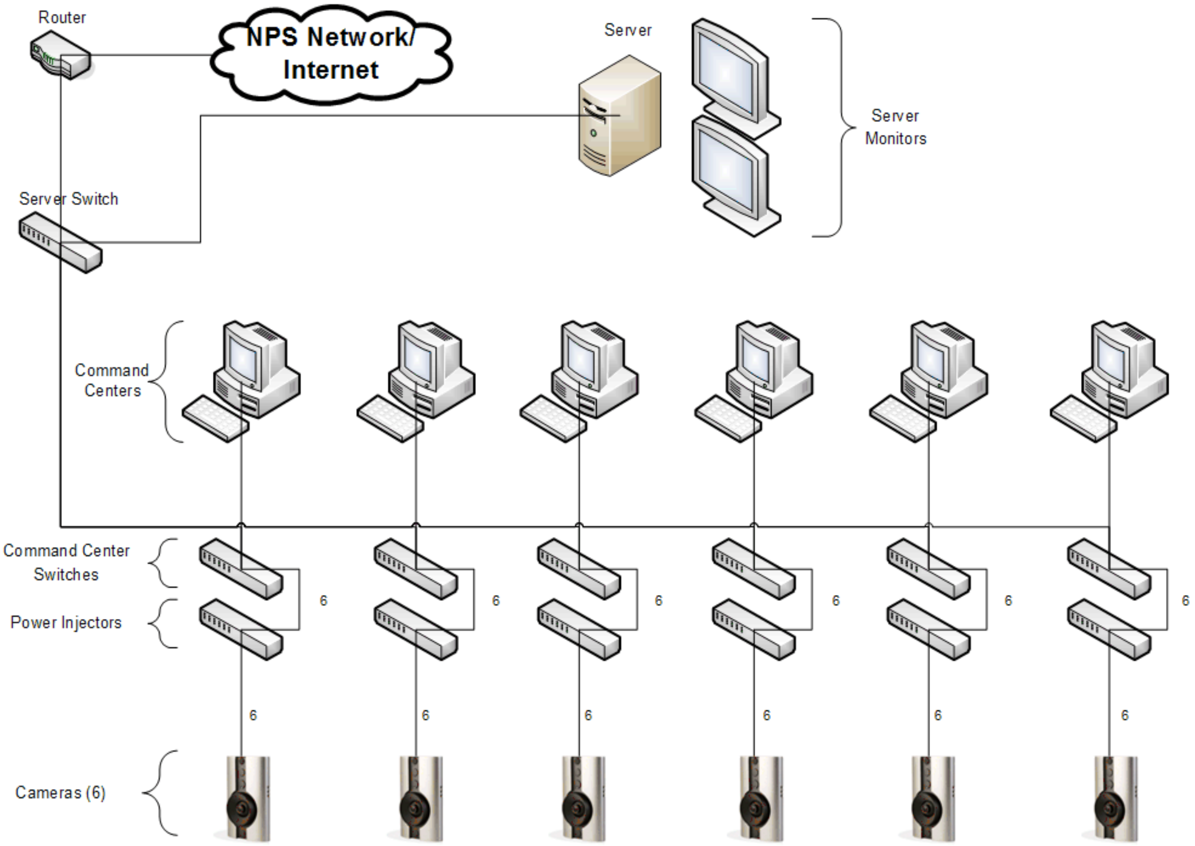


Figure 52. Network Subsystem Topology

(1) Computer Placement. The command centers, switches, and server are installed on the workbench in BU201L. This location provides convenience and security.

(2) Cable Routing. Cables connecting the switches, "power injectors," "command centers," and server are arranged within the workbench in BU-201L. Cables connecting the "power injectors" to the individual cameras are routed through the existing floor conduit or upward to the existing pipe hangers and outward toward the cameras. Cables follow the conduit or pipe hangers as much as possible and are securely and aesthetically routed along the ceiling otherwise. Cables are secured with plastic tie-wraps. Cable lengths vary between 30 feet and 150 feet and none are greater than the maximum length of 300 feet.

(3) Cable Construction. Cables are made from a bulk Cat 5e cable spool and standard 8P8C connectors wired to T568B standards. The cable lengths are adequate to attach the respective camera to its switch port plus ten feet of cable in reserve to allow for minor hardware relocation or changes to cable routing paths, if required.

c. Logical Network Topology

Due to the interconnection of multiple camera groups with multiple command centers, each network host is assigned a static IP address. Camera IP addresses are set from their respective command centers using the vendor-provided utility program. The addressing scheme adheres to the following pattern:

- All addresses use the 192.168.0.x class C private IP address space with a network mask of 255.255.255.0.
- The Apple server uses the address 192.168.0.100
- Each command center uses the address 192.168.0.X0, where X corresponds to the number of that command center. (e.g., command center #3 will use 192.168.0.30)

- Each camera uses the address 192.168.0.XY, where X corresponds to the number of the command center and Y corresponds to the number of the camera.
- The router uses the address 192.168.0.1. This means that the server and all command centers will use 192.168.0.1 as their default gateway.

d. Network Engineering Subsystem Detailed Engineering Specifications

This subsection itemizes the detailed engineering specifications for the WNES, listed as enumerated system requirements.

(1) Requirement 6.0: Hardware List

The camera network consists of the following hardware:

- 6, WiLife Master Camera Systems with Command Center Software WLPIC-4X6
- 10, WLIR-50 NightVision Illuminator Kits
- 10, WiLife WLAL-120 120 degree wide angle lens for indoor camera
- 10, WiLife WLAL-54 54 degree close up lens for indoor cameras
- 10, QuickCam Pro 9000 USB 2.0 Webcam
- 6, D-Link 8-port 10/100Mbps switches
- 1, Linksys 8-port 10/100Mbps switch
- 1, Linksys WRT54G router
- 3500 feet of CAT5e cabling with 8P8C connectors
- 6 DELL Precision Workstations, T3400 (375/32bit)

(2) Requirement 7.0: Camera Placement

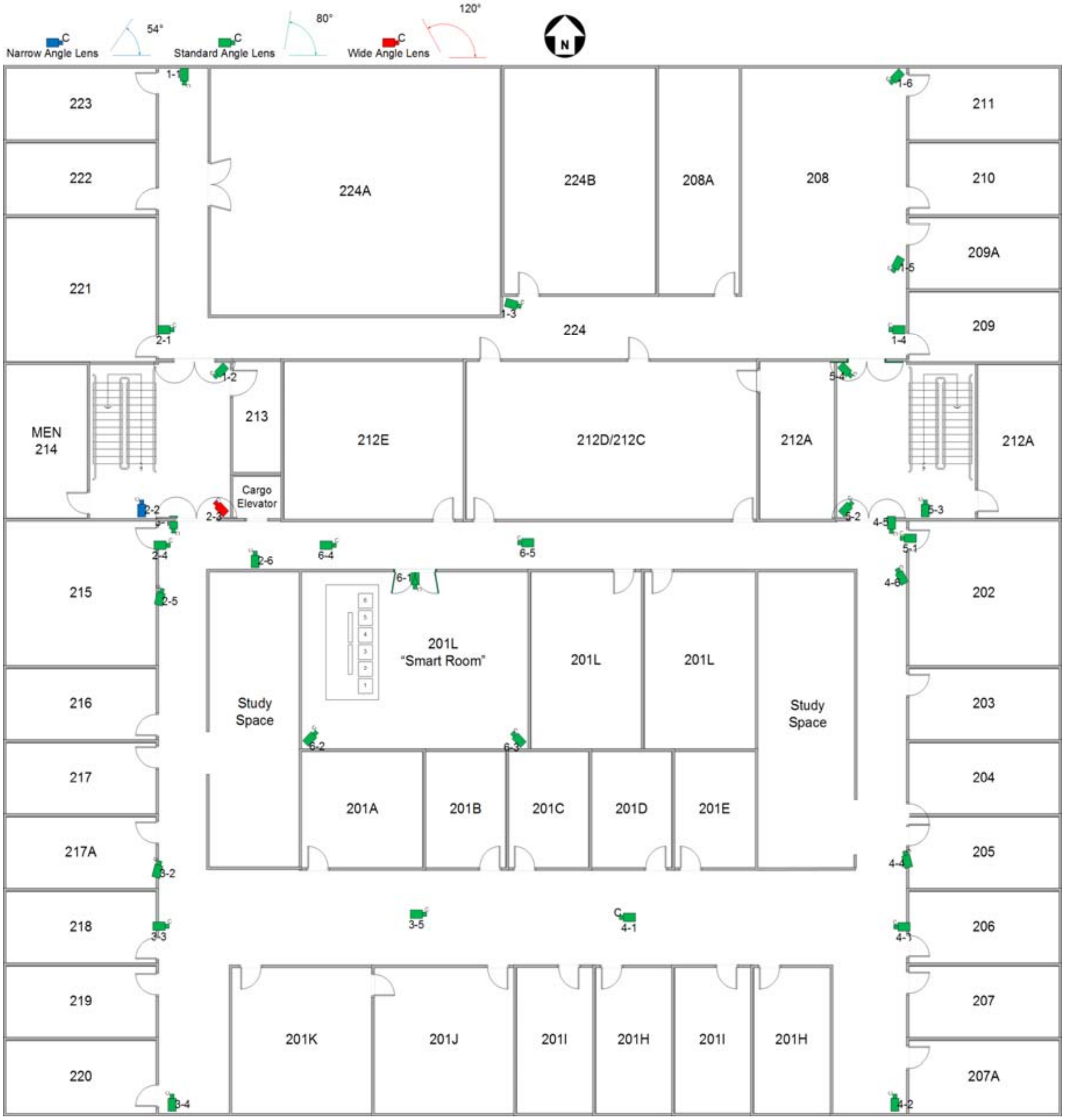


Figure 53. Camera and Network Hardware Placement

Figure 53 graphically depicts camera and network hardware placement. Figures 54 through 59 show each camera group (with up to six cameras each) with the approximate field of view from each camera and the cable routing from the equipment cluster to each camera. Cameras have been placed such that most areas will be visible by at least two cameras. Cameras are grouped such that every group has some coverage overlap with at least one other group. Stairwell areas and the hallway across from the freight elevator use camera placement optimized for a close-up view of the faces of people entering the area.

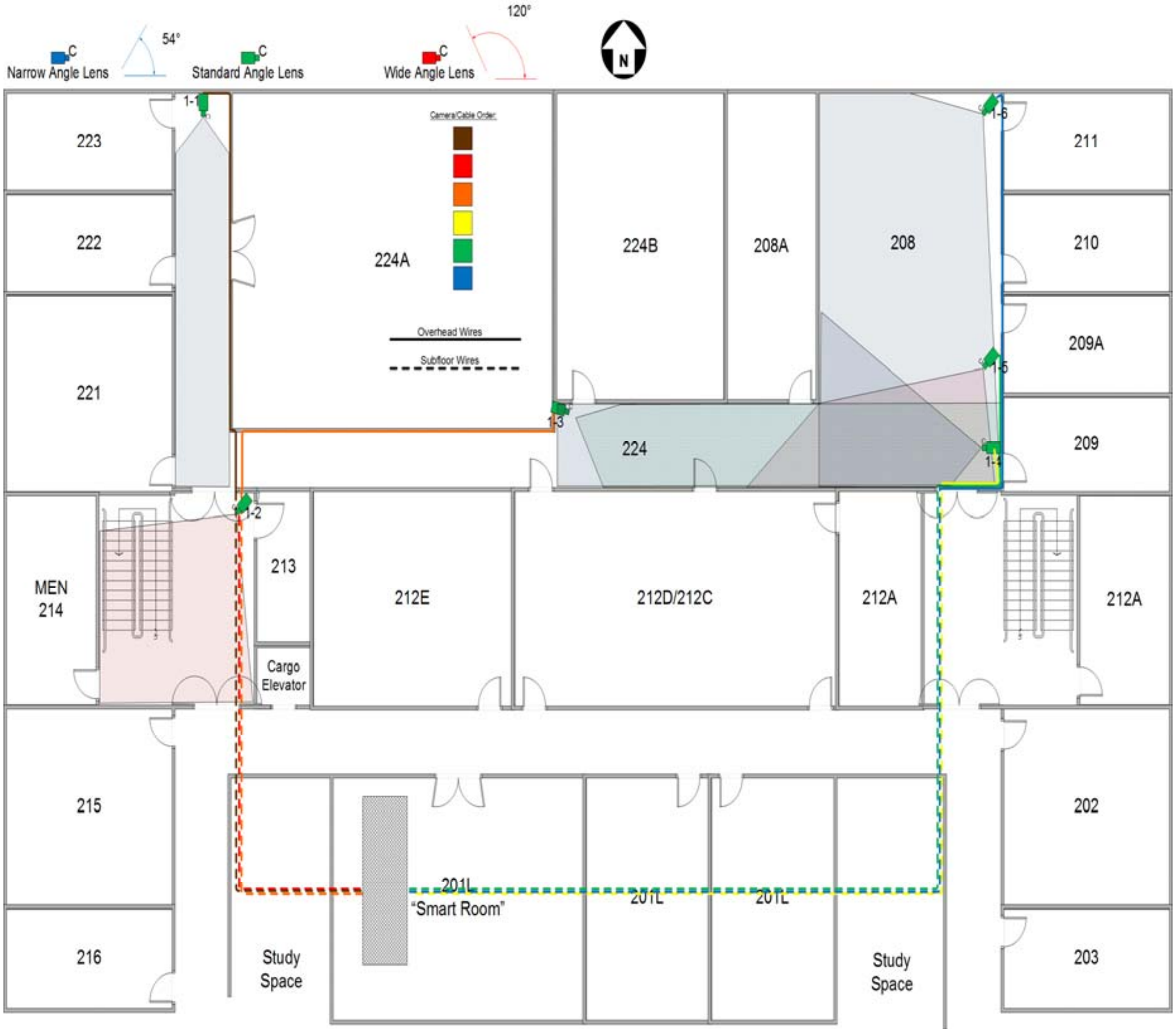


Figure 54. Camera Group 1

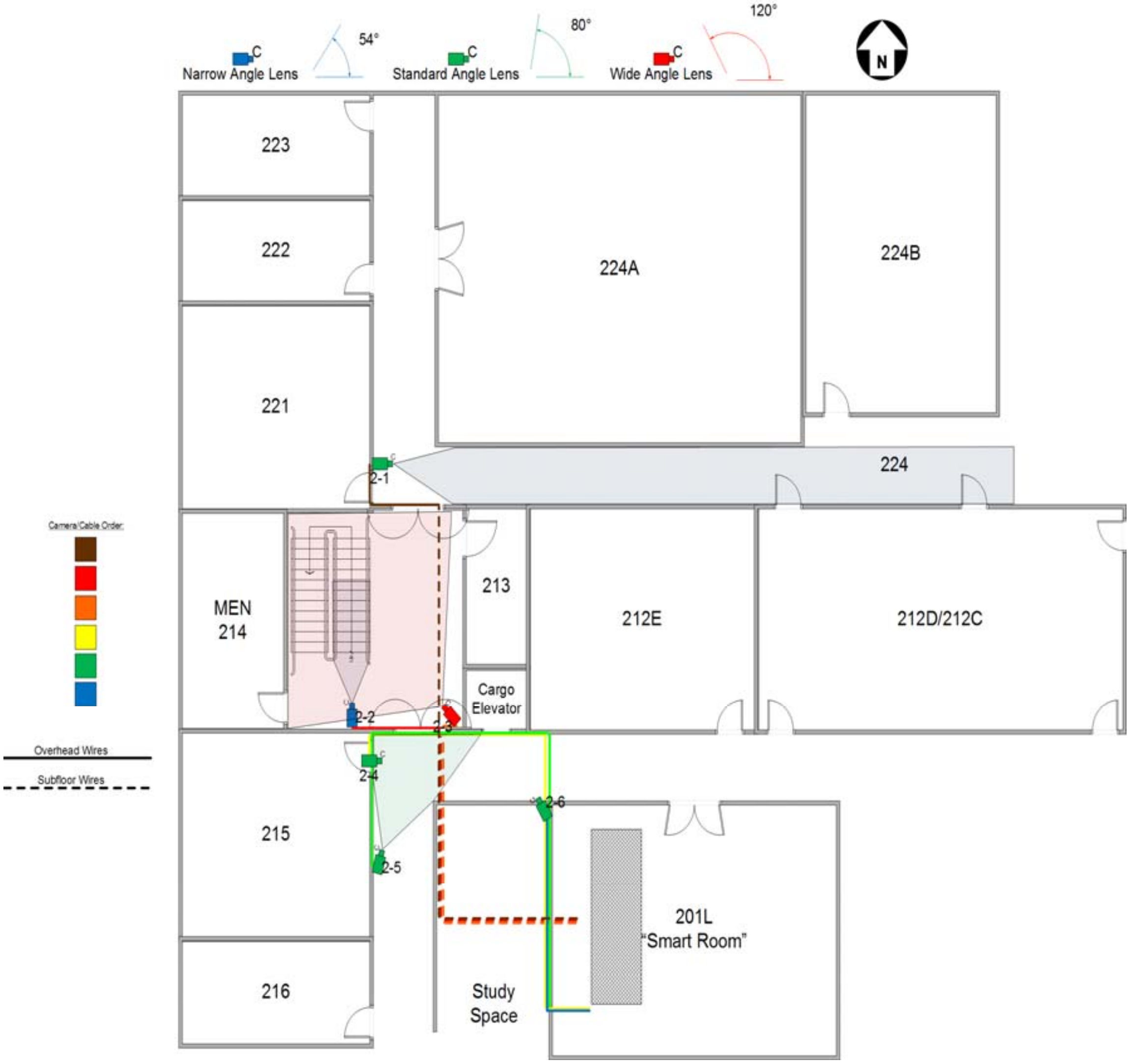


Figure 55. Camera Group 2

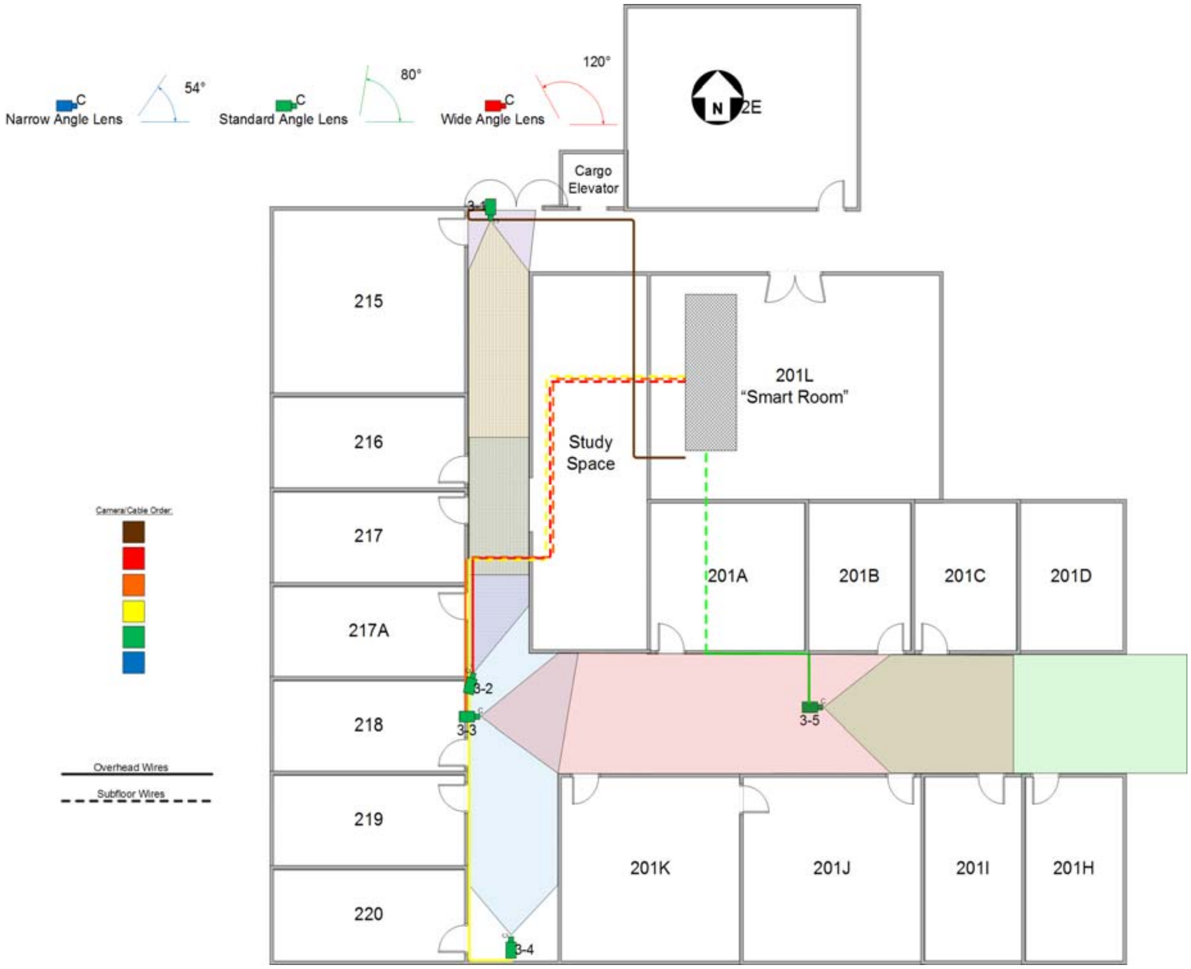


Figure 56. Camera Group 3

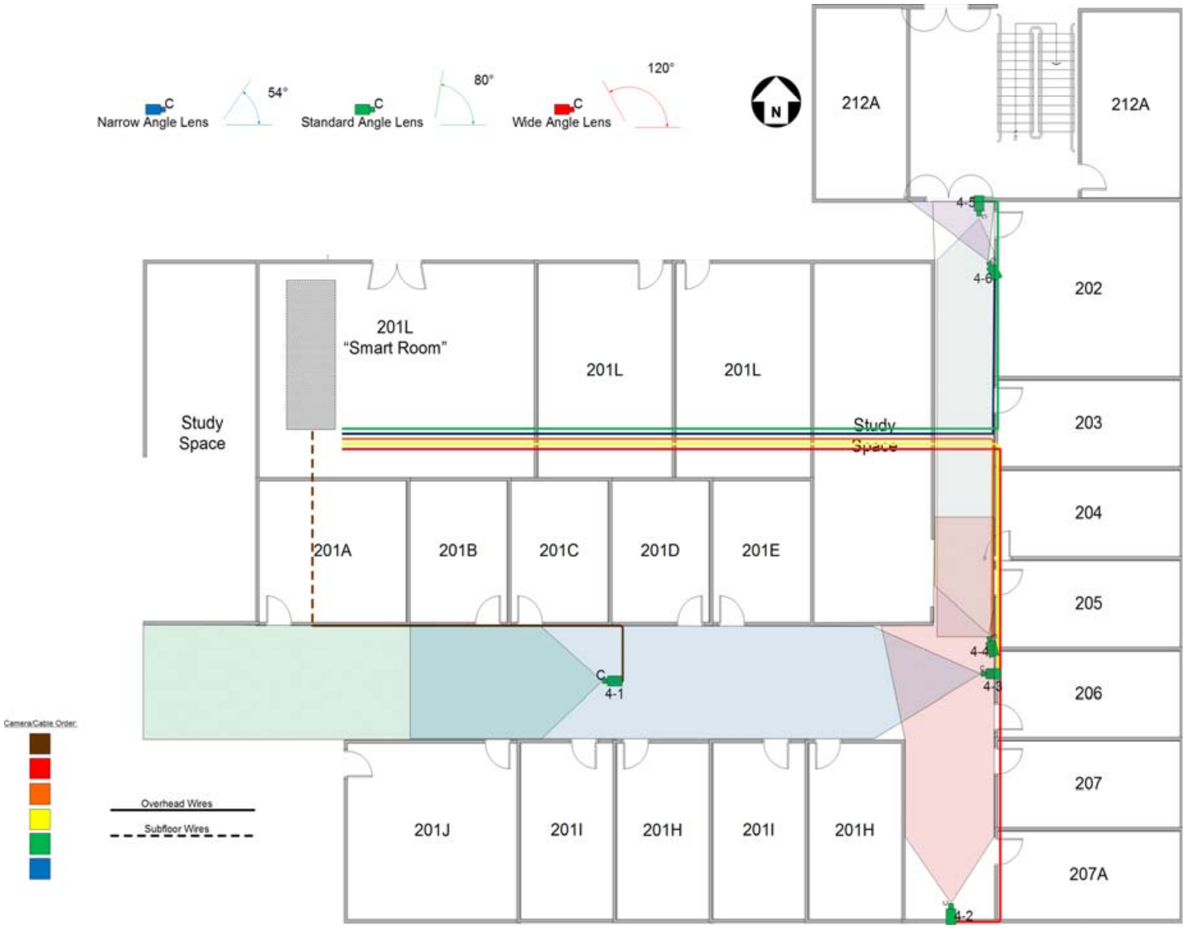


Figure 57. Camera Group 4

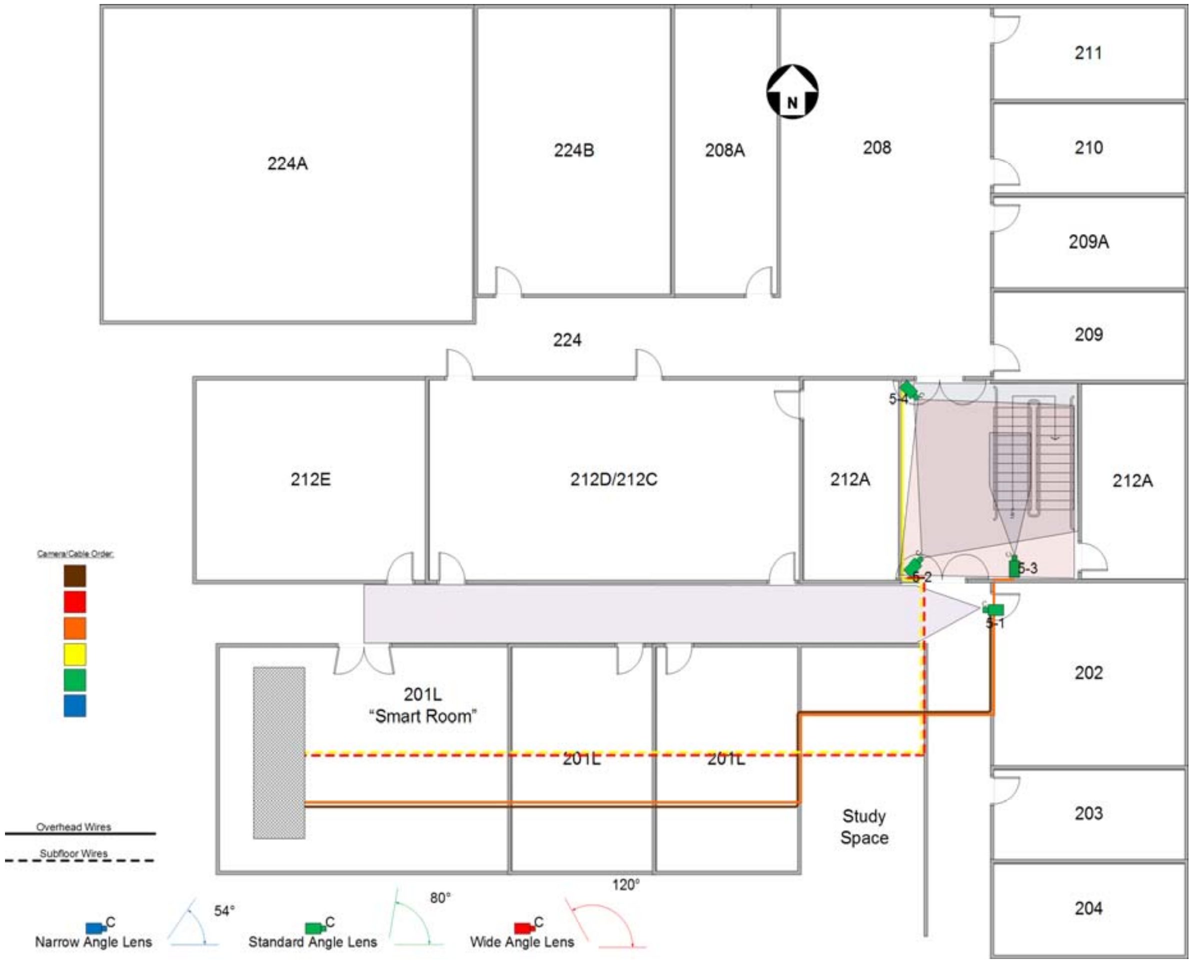


Figure 58. Camera Group 5

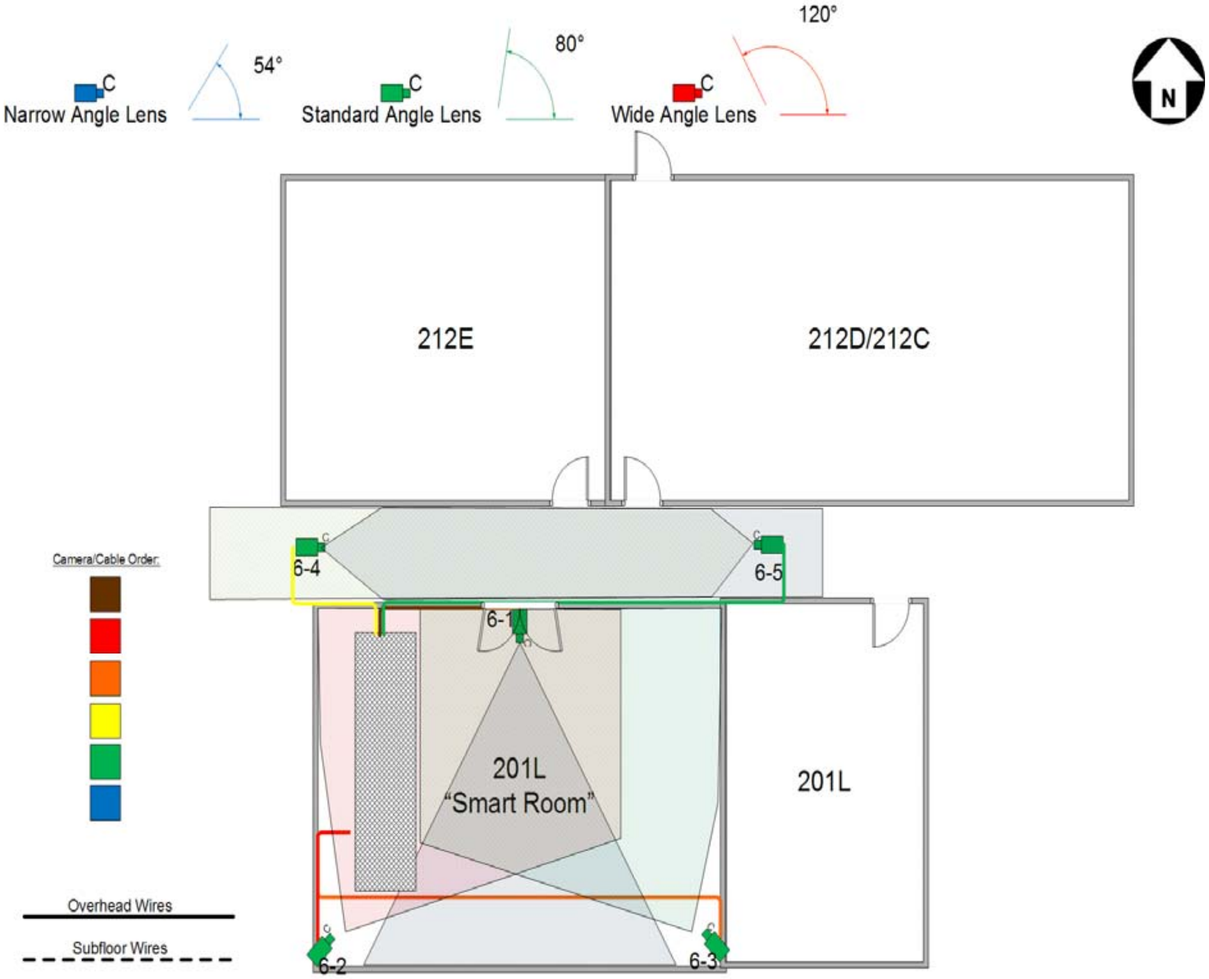


Figure 59. Camera Group 6

(3) Requirement 8.0: Camera Installation

Cameras are fastened to 4-inch square pieces of thin Plexiglas using the included mounting hardware. This Plexiglas mounting plate uses adhesive-backed Velcro to attach to the wall mounting points.

(4) Requirement 9.0: Data Rate Capacity

Network shall provide stable high-speed (at least 100Mbps) data connectivity for the following data transfer requirements:

- Transfer of Video Data
- Transfer of Processed Contact Data

- Transfer of Behavior Classification Data
- Timely Generation of Alerts

e. Subsystem Acceptance Test

The full subsystem acceptance test was accomplished to complete the verification step of the SE Process by verifying that all components of the system were properly functioning and communicating with each other. The following test procedure is included to describe the verification steps, which were taken for this subsystem of the entire system. This step was a key component that led to the overall verification and validation during the full system acceptance test procedure.

1. Power test:

    - Power up the server, command centers, switches, and cameras and verify proper indications.
    - The server and command centers should exhibit a normal power-on self-test sequence.
    - The switches should show power LED lit.
    - The cameras should show the network connectivity LED lit at its port on its switch.

2. Connectivity test:

    - From the server, issue a ping command to each network adapter.
    - Each command center and camera should return a successful ping reply.

3. Video data test:

    - Start the WiLife application on each command center to verify the reception of video data from each camera.
    - Each command center should display a video image from each attached camera.

4. Hardware installation review:

- Inspect all hardware, including network cabling, to verify that hardware is properly, securely, and aesthetically installed.
- Computers should be stable, have efficient cable routing, and have adequate airflow. Keyboards and monitors should be ergonomically operable from the expected user stations.
- Network cables should be securely fastened to walls and structures. They should not present a safety hazard (trip hazard or entanglement). They should present an aesthetic and professional appearance.

Cameras should be securely mounted to their attachment points.

4. Software Engineering Subsystem

This subsection describes the Software Engineering Subsystem within the WMSE POCS.

a. Software Engineering Subsystem Overview

The Watchman Software Engineering Subsystem (WSES) (Figure 60) receives raw video data input from the WiLife DLC-810i indoor cameras. Data transfer is provided by the WNES. The detection and identification software at each PC node receives this data for processing. The WSES processes the video input to extract meaningful detection and identification data. This data is aggregated and stored in the WSES database and the WAES databases for access by the server for higher-level processing by the WAES. The data stored in the subsystem databases originally included:

- Unique Identifier (UID): unique identifier.
- Date: System date at processing.
- Time: System time at processing.

- Node (N#): PC node number.
- Camera (C#): Camera number.
- Field of View (FOV) vector containing:
- Area: Area of blob detected
- Histogram: Histogram of blob detected.
- Face Detected: True or False.
- Figure of Merit (FOM): calculated FOM using FOV and FR data
- ID: Identification from Personnel Database.
- Confidence Factor (CF): Facial Recognition match percentile for ID.

The WSES was initially be designed to process detection and identification for one (1) person at a time to provide 80% facial recognition accuracy for identification and mustering and 90% detection for behavior analysis. These performance metrics were ultimately derived from the operational user requirements under the DRM, which are the MDA JIC capability gaps. The specific gaps that these metrics address is MDA 004C-002T/003T, "Identify adversary patterns of behavior," and "Differentiate maritime threats from valid maritime commerce." The facial recognition metric is a measure of achieving the differentiation need, while the behavior analysis metric measures achievement of the behavior pattern recognition need. Access to archived video data will be available to the server via the network subsystem. The camera video data is stored and managed by the WiLife Command Center to allow protection and on-demand access by the server. Additionally, continuous real-time streaming video data is available to the server via the network subsystem.

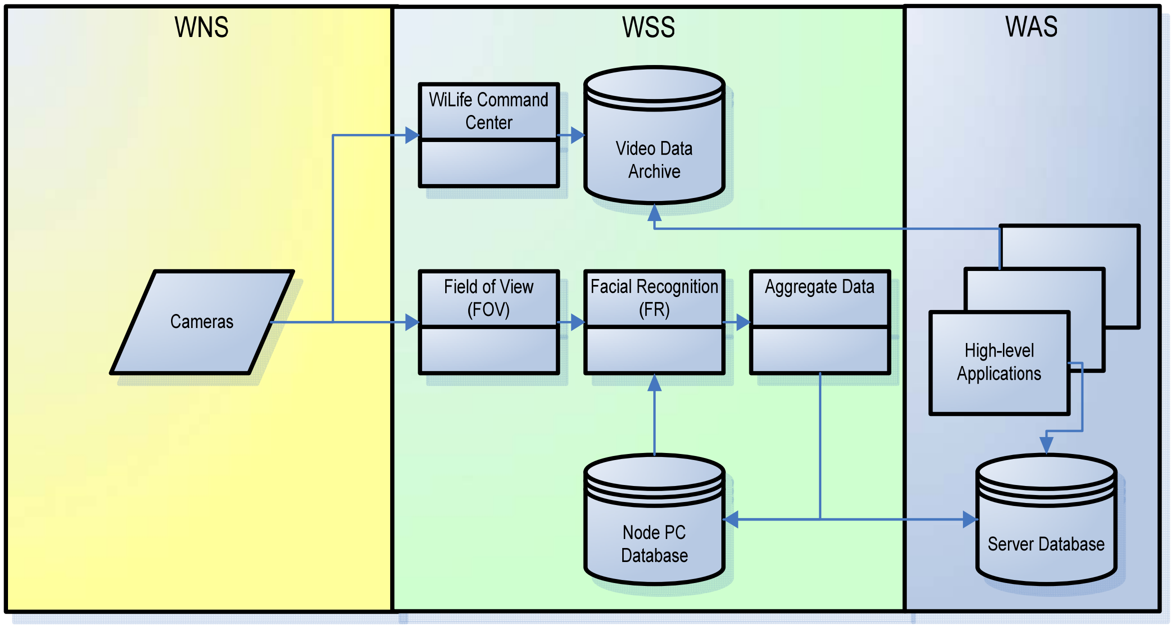


Figure 60. Watchman Software Subsystem (WSS) overview

b. *List of Software*

The Software Subsystem utilizes both commercial off the shelf (COTS) and locally generated software for implementation. This software was required to be loaded on each Node Personal Computer (PC) in order to perform the required processing.

(1) Commercial off the Shelf (COTS) Software. The COTS software utilized within this subsystem consists of the following components with their requisite purposes:

- WiLife Command Center: camera setup and video archive management.
- MATLAB
    - Blob Analysis.
    - Histogram.
    - Face Detection.
    - Face Recognition.

- MS SQL Server Express: data warehouse.

Software requirements not met using COTS software were locally generated to capture, create, and store data in the databases and perform several other housekeeping functions that were required.

(2) Software Installation. The Node PCs are running the Windows XP operating System. NPS Information Technology and Communications Services (ITACS) personnel loaded MATLAB 2008a, MS Office and MS SQL Server Express on each Node PC. Additional software has been and can continue to be loaded on the Node PCs and WAES Mac Server by the WSES team as required, due to the open architecture design of the system.

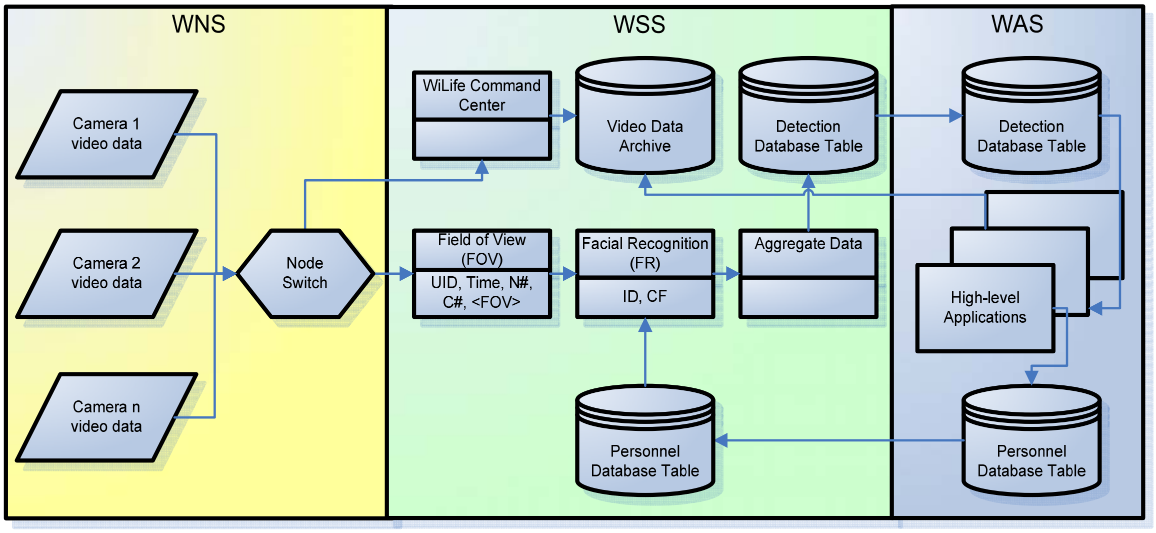


Figure 61. Watchman Software Engineering Subsystem (WSES) block diagram

c. *Software Engineering Subsystem Detailed Engineering Specifications*

This subsection itemizes the detailed engineering specifications for the WSES, listed as enumerated system requirements.

(1) Requirement 10.0: WiLife DLC-810i Indoor Cameras. Video input will be received from the WiLife cameras via the WNS for storage and processing within the WSS. WiLife camera specifications are listing in Table 9.

Processing Power:	400 MHz DSP
Onboard Image Encoding:	Windows Media Video 9
Onboard Image Processing:	Motion detection up to 16 zones, auto-brightness
Resolution:	320 x 240 or 640 x 480 pixels
Frame Rate:	5, 10 or 15 frames per second,/font>
Bit Rate:	150 kbs to 800 kbs,/font>
Color Depth:	10 bits
Focus:	User-adjustable
Viewing Angle:	80° (diagonal)
Pan Angle:	50° (manual)
Firmware Updates:	Manual or Automatic
Communications:	Ethernet, TCP/IP
Power Consumption:	15 Watts

Table 9. WiLife DLC-810i Indoor Cameras Specifications

(2) Requirement 11.0: WiLife Command Center. The WiLife Command Center software was installed on each node PC and automatically stores the video input in Windows Media Player 9 (.wmv) format files on the local hard drive when the cameras detect motion. The software will also organize the files and manage the allocated video storage. Hard drive storage space is allocated for video files within the software and when the video files fill up the allocated space the software automatically removes the oldest files to make room for new recordings. Each Node PC must remain on for the system to work and record video.

The software interface is user-friendly. Monitoring of cameras or playback of archived video clips is possible from any or all of the cameras connected to each Node PC. Additionally, the software allows for saving video clips and still frame images (jpg, gif, png, tiff, etc). The system can easily be configured to automatically send an e-mail or cell phone alert with a video clip or still frame when motion is detected, though this functionality was not enabled due to lack of defined requirement.

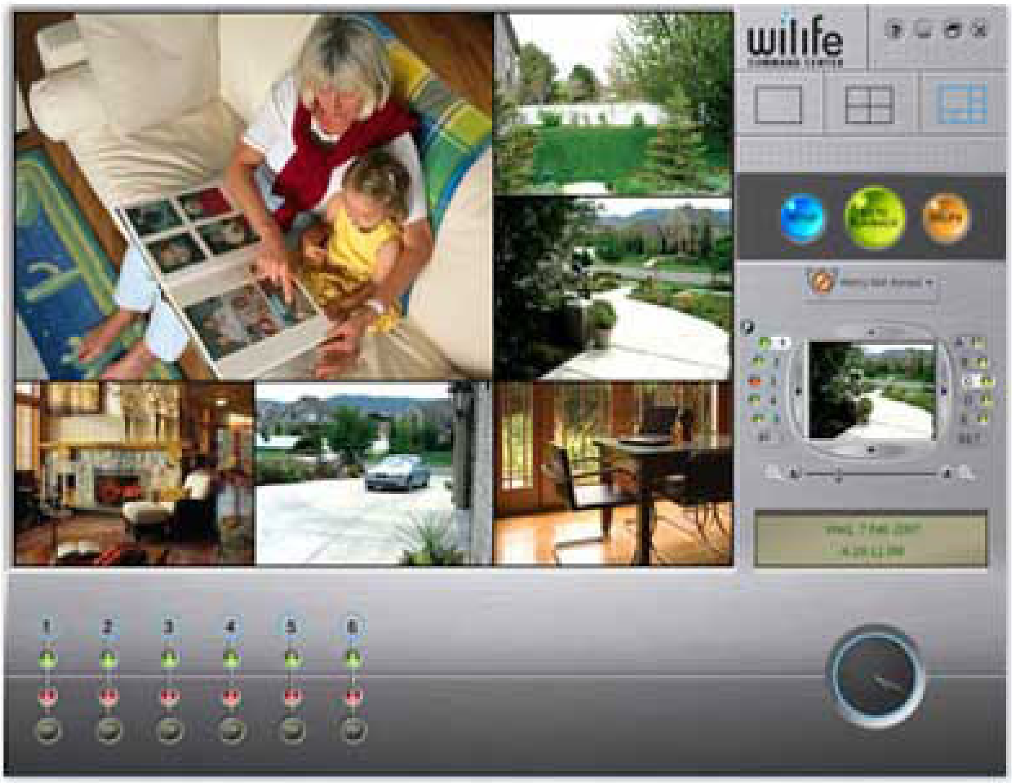


Figure 62. WiLife Command Center Software screen capture.

i. Requirement 12.0: Field of View (FOV)

- Function: Analyze node camera live video inputs to calculate the FOM vector data for each camera's input that contains; object area, color histogram, face detected, and FOM.
- Inputs: Live video feed in Windows Media Video 9 format from all node cameras.
- Output: UID, Date, Time, Node#, C#, Area, Color Histogram, Face Detected, and FOM.
- Software: MATLAB.

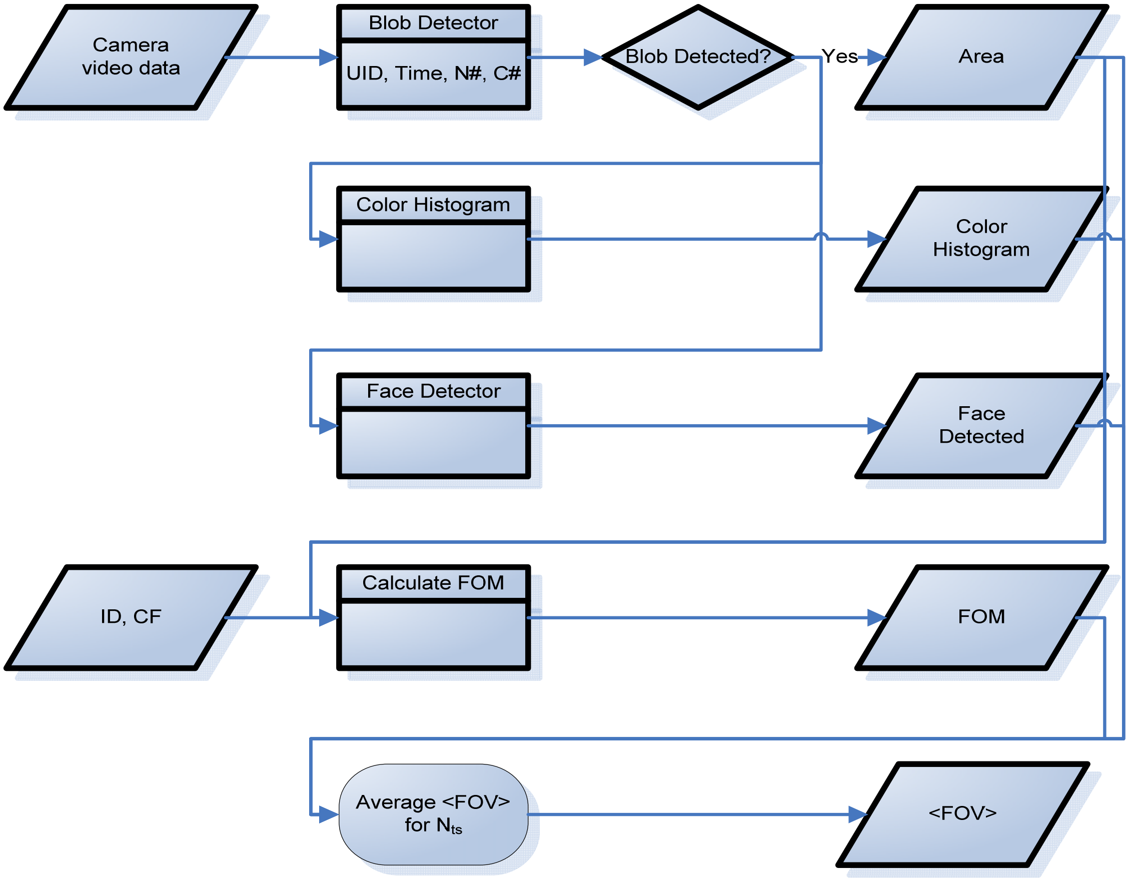


Figure 63. Field of View (FOV) data flow chart

ii. Requirement 13.0: Facial Recognition (FR)

- Function: Given a live video feed with a properly positioned camera provide 80% facial recognition accuracy.
- Input: Database of Faces with Names, Live Windows Media Video 9 (WMV) video feed output from FOV if a face is detected.
- Output: Potential Match with UID, Time, Node#, C#, ID, and Confidence Factor (CF) of match.
- Software: MATLAB.

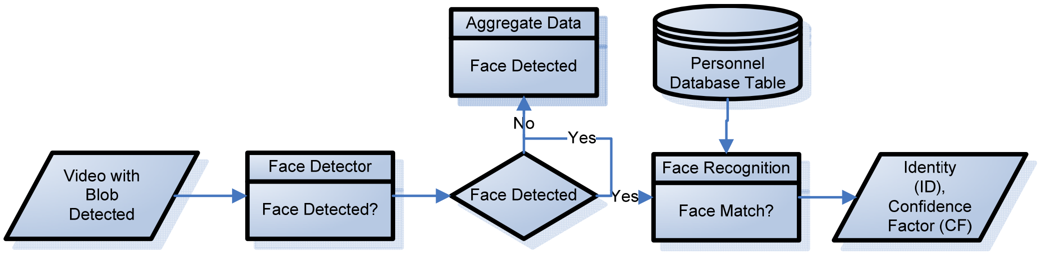


Figure 64. Face Recognition (FR) data flow chart

iii. Requirement 14.0: Aggregate Data

- Function: Filter data using a precedence scheme and Network Time Scale (N_{TS}) to store the best FOV and FR data in the databases on the server and Node PCs.
- Input: FOV and FR data.
- Output: "Best" data over a set time interval to server and local database in a given format for the data information package: [UID, Time, N#, C#, Area, Color Histogram, Face Detected, FOM, ID, CF].
- Software: MATLAB.

iv. Requirement 15.0: Data Warehouse

The data storage requirements are accomplished using a database within each Node PC as well as an aggregate database on the WAES server that will contain the data from each Node PC. The personnel data for each individual will be stored in the database in the table named *Personnel*. The detection data will be stored in the database in the table named *Detection*.

v. Requirement 16.0: Detection Table

- Function: Store Facial views for usage by facial recognition software.
- Input: Multiple Facial Profiles for a Given Test Group.
- Output: Pass Data as called for by Facial Recognition Software.
- Software: MS SQL Server Express.

Data Dictionary

Table name: Detection

Primary Key: UID

Field	**Type**	**Nulls?**
UID	int	No
Time	date/time	No
Node (N#)	int	No
Camera (C#)	int	No
ID	int	Yes
FR_CF	percentage	Yes
Height	double	No
Width	double	No
Color	double	No
FOM	double	No

Index Name	**Unique**	**Clustered**	**Fields**
PK_Detection	Yes	Yes	UID
Time	No	No	Time
Camera	No	No	Camera

Internal Foreign Key Constraint	**Affected Field**	**Source Table**
FK_Detection_Personnel	PersID	Personnel

Table 10. Detection Table from the Detection Database

vi. Requirement 17.0: Detection Database

- Function: Store data to allow for behavior analysis and mustering.
- Input: Data fields from the aggregate data software: [UID, Time, N#, C#, Area, Color Histogram, Face Detected, FOM, ID, CF]
- Output: Data to behavioral analysis software at server for processing.
- Software: MS Access

Data Dictionary

Table name: Personnel

Primary Key: PersID

Field	**Type**	**Nulls?**
PersID	int	No
First_Name	text (15)	No
M_I	text (1)	No
Last_Name	text (25)	No
Email	text (25)	No

Index Name	**Unique**	**Clustered**	**Fields**
PK_Personnel	Yes	No	PersID

Primary Key as Foreign Key Constraint	**Affected Table**	**Affected Field**
FK_Personnel_Detection	Detection	ID

Table 11. Personnel Table from the Detection Database

vii. Requirement 18.0: Video Data Store

- Function: Provide the server access to video data files saved on the Node PCs by the WiLife Command Center software. Allow data files to be protected to prevent automatic overwrite.
- Input: Recorded video in Windows Media Player 9 format from WiLife Command Center, Protection requests.
- Output: Saved video file in Windows Media Player 9 format on local hard drive of each Node PC.
- Software: WiLife COTS Software package

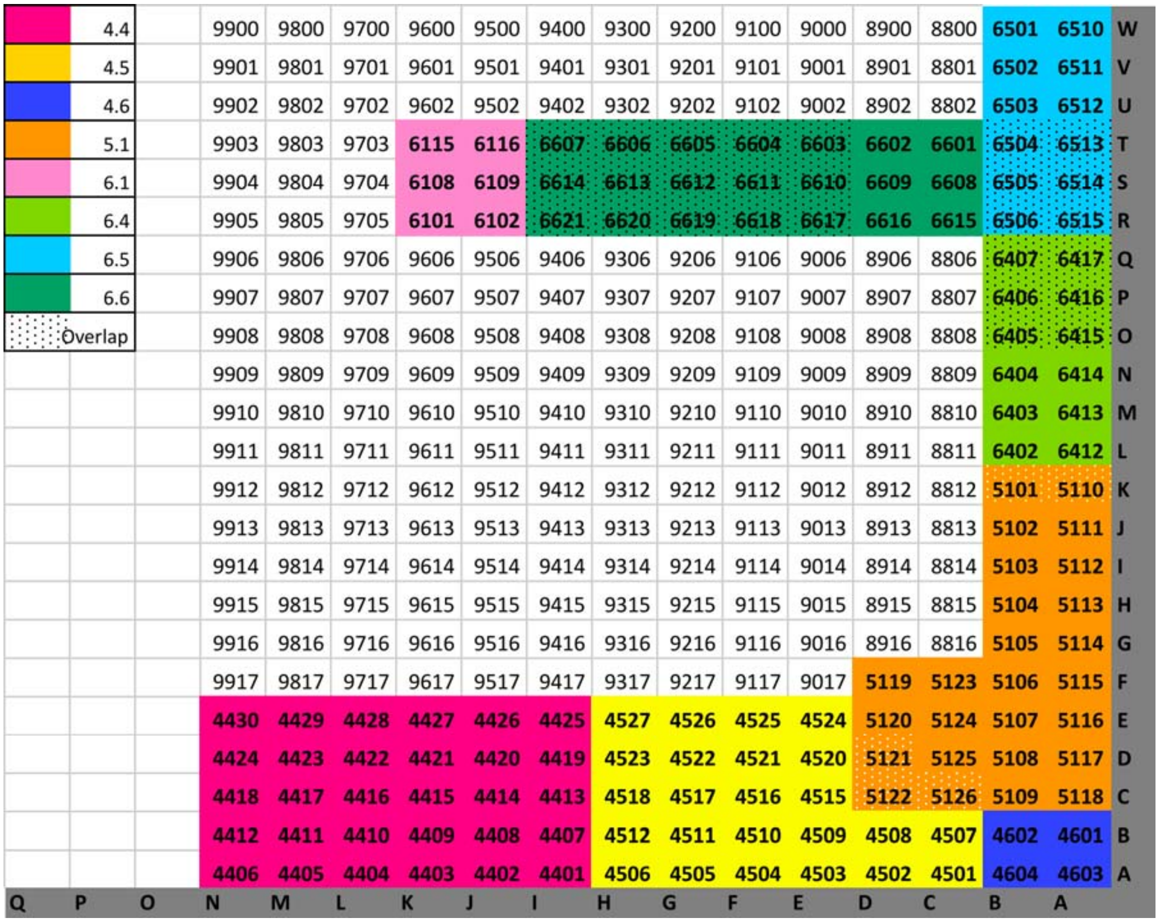
5. Upgrades to the Initial Watchman System

This section will describe some of the significant Engineering Change Orders (ECOs) which were implemented in enhance the base capability of the Watchman Maritime Smart Environment (WMSE) concept development and demonstration system, as well as additional system elements which were integrated into the original Watchman system to expand the capability of the POCS. The following section of this chapter will discuss how these enhancements and expansions affect the recommended method for operational implementation of the system.

a. Engineering Change Orders (ECOs)

(1) Update Behavior Analysis Code. This ECO was required to accomplish the upgrading and enhancing of the Behavior Analysis Module (BAM) of the Watchman System to implement additional aspects of functionality that was not implemented in the first quarter phase of the project. Due to the anticipated impact to the interfaces of the BAM with other system elements, e.g., the Position Integration Module (PIM), the server, and the Detection database, the BAM code had to be updated to ensure that not only did the BAM utilize the increased functionality derived from the other ECOs, but also that the BAM still functioned as designed, given the expected interface changes.

In particular, Watchman ECO 3, Camera Calibration, enhanced the system's ability to more precisely locate the tracked target in the surveillance area. This enhanced position information was provided to the BAM, which then had to process this data in order to capitalize on the greater fidelity of behavior pattern recognition that the higher position resolution provided. Figure 65 shows the mapping of the "Zones" which were created via the camera calibration procedure, which were in turn translated into a text file that was then read into the BAM for precise behavior sequence processing.



🟥	4.4		9900	9800	9700	9600	9500	9400	9300	9200	9100	9000	8900	8800	6501	6510	W
🟨	4.5		9901	9801	9701	9601	9501	9401	9301	9201	9101	9001	8901	8801	6502	6511	V
🟦	4.6		9902	9802	9702	9602	9502	9402	9302	9202	9102	9002	8902	8802	6503	6512	U
🟧	5.1		9903	9803	9703	6115	6116	6607	6606	6605	6604	6603	6602	6601	6504	6513	T
🟪	6.1		9904	9804	9704	6108	6109	6614	6613	6612	6611	6610	6609	6608	6505	6514	S
🟩	6.4		9905	9805	9705	6101	6102	6621	6620	6619	6618	6617	6616	6615	6506	6515	R
🟦	6.5		9906	9806	9706	9606	9506	9406	9306	9206	9106	9006	8906	8806	6407	6417	Q
🟩	6.6		9907	9807	9707	9607	9507	9407	9307	9207	9107	9007	8907	8807	6406	6416	P
▨	Overlap		9908	9808	9708	9608	9508	9408	9308	9208	9108	9008	8908	8808	6405	6415	O
			9909	9809	9709	9609	9509	9409	9309	9209	9109	9009	8909	8809	6404	6414	N
			9910	9810	9710	9610	9510	9410	9310	9210	9110	9010	8910	8810	6403	6413	M
			9911	9811	9711	9611	9511	9411	9311	9211	9111	9011	8911	8811	6402	6412	L
			9912	9812	9712	9612	9512	9412	9312	9212	9112	9012	8912	8812	5101	5110	K
			9913	9813	9713	9613	9513	9413	9313	9213	9113	9013	8913	8813	5102	5111	J
			9914	9814	9714	9614	9514	9414	9314	9214	9114	9014	8914	8814	5103	5112	I
			9915	9815	9715	9615	9515	9415	9315	9215	9115	9015	8915	8815	5104	5113	H
			9916	9816	9716	9616	9516	9416	9316	9216	9116	9016	8916	8816	5105	5114	G
			9917	9817	9717	9617	9517	9417	9317	9217	9117	9017	5119	5123	5106	5115	F
			4430	4429	4428	4427	4426	4425	4527	4526	4525	4524	5120	5124	5107	5116	E
			4424	4423	4422	4421	4420	4419	4523	4522	4521	4520	5121	5125	5108	5117	D
			4418	4417	4416	4415	4414	4413	4518	4517	4516	4515	5122	5126	5109	5118	C
			4412	4411	4410	4409	4408	4407	4512	4511	4510	4509	4508	4507	4602	4601	B
			4406	4405	4404	4403	4402	4401	4506	4505	4504	4503	4502	4501	4604	4603	A
Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	

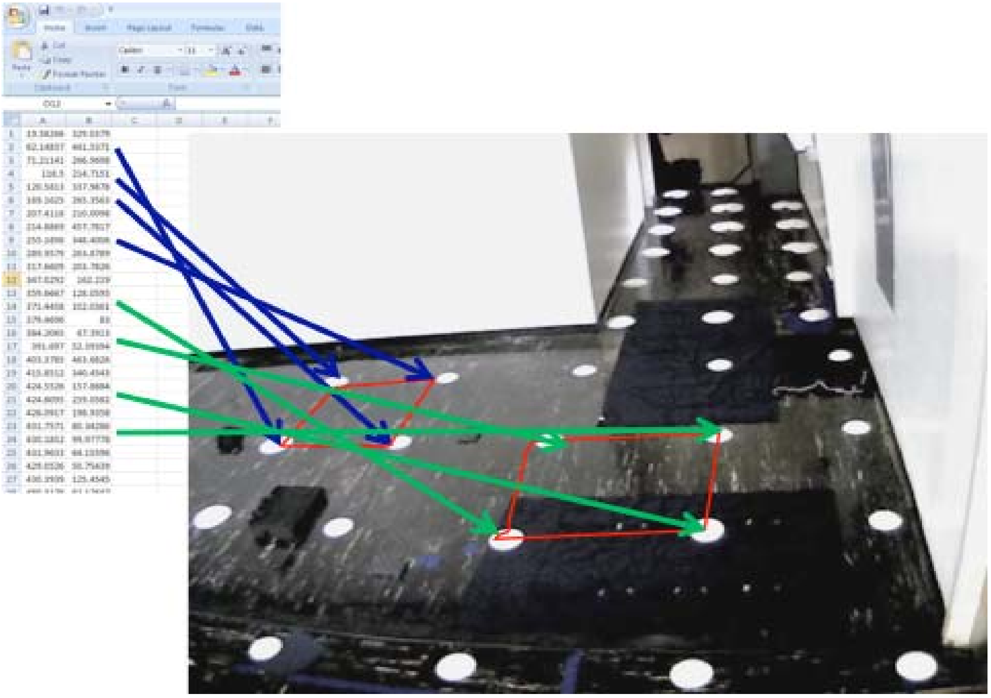
Figure 65. Camera Calibration "Zone Mapping"

(2) Camera Calibration. The Camera Calibration ECO was implemented to correct the Watchman System's inability to precisely determine the position of a track within the Watchman System's surveillance area. The original Watchman configuration gave only a general location for the track, this general location being the field of view of the system camera that had the largest image (blob size) of the track, and therefore the highest FOM. The original general location varied from a 400 square-foot (sq-ft) area (the Watchman Lab) to a 200 sq-ft hallway to a 60 sq-ft stairwell.

To provide the ability to analyze more discrete behaviors, the system had to be able to provide a refined, more accurate location of the target (in this case, a person) being tracked. In other words, an enhanced position detection function.

This refined position detection allowed the Watchman System to more accurately capture the movements of the track as it moved within the surveillance area. This afforded the added advantage of permitting the system to have more precise behaviors identified, documented, and captured in the BAM, thus allowing the system to be more agile and effective at recognizing more detailed Normal, Abnormal and Unknown behaviors.

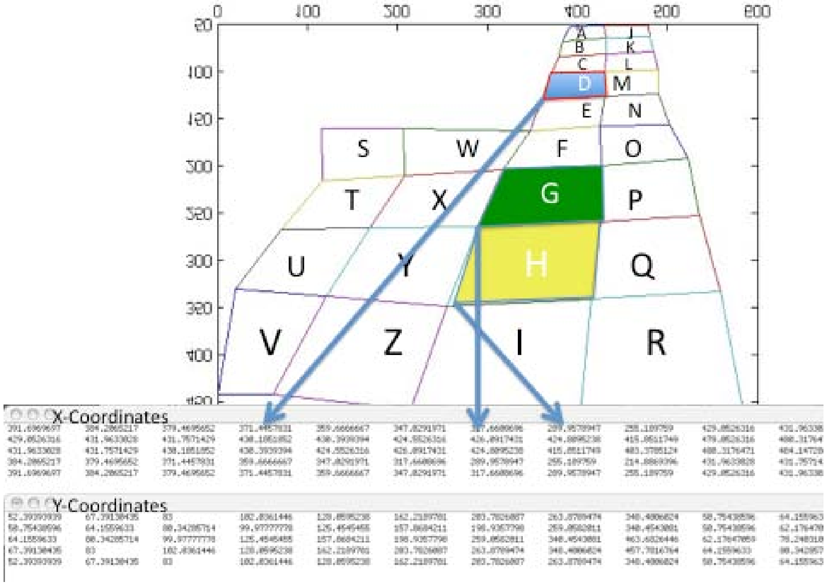
The initial calibration procedure was a very labor-intensive process of mapping out each camera's FOV with markers to identify the zones, and then laboriously manually processing the resulting image to allow the calibration program from MATLAB to automatically scan and identify the markers from within the image, as shown in Figure 66, then write them to a file.



	A	B
1	13.58266	329.0379
2	62.14837	461.3371
3	71.21141	266.9698
4	128.5	234.7151
5	120.5813	337.9878
6	169.1625	265.3563
7	207.4118	216.0098
8	214.8869	457.7817
9	255.1898	348.4058
10	289.9579	264.8789
11	317.6609	203.7826
12	347.0292	162.229
13	359.6667	128.0593
14	371.4458	102.0381
15	379.4698	83
16	384.2093	67.3913
17	391.497	52.39184
18	403.3783	461.6826
19	413.8512	340.4343
20	424.5526	157.8884
21	424.8095	239.0582
22	426.0917	198.9358
23	431.7571	80.38288
24	430.3852	99.97778
25	431.9631	64.13398
26	429.0526	50.75639
27	430.3939	125.4545

Figure 66. Camera Calibration Image Processing

These marker positions, as identified by the MATLAB automatic image scan, then had to be manually cross-referenced and re-ordered into the actual zones as they appear in the camera FOV to the human eye perspective, by creating matrices of the (X,Y) coordinates for all the marker positions. To ensure that all the zones were correctly entered into the matrices, a figure of the polygons using MATLAB had to be plotted. The matrices were saved in text files that could be read in by a MATLAB function to evaluate if a blob's position is inside any of the zones.



X-Coordinates

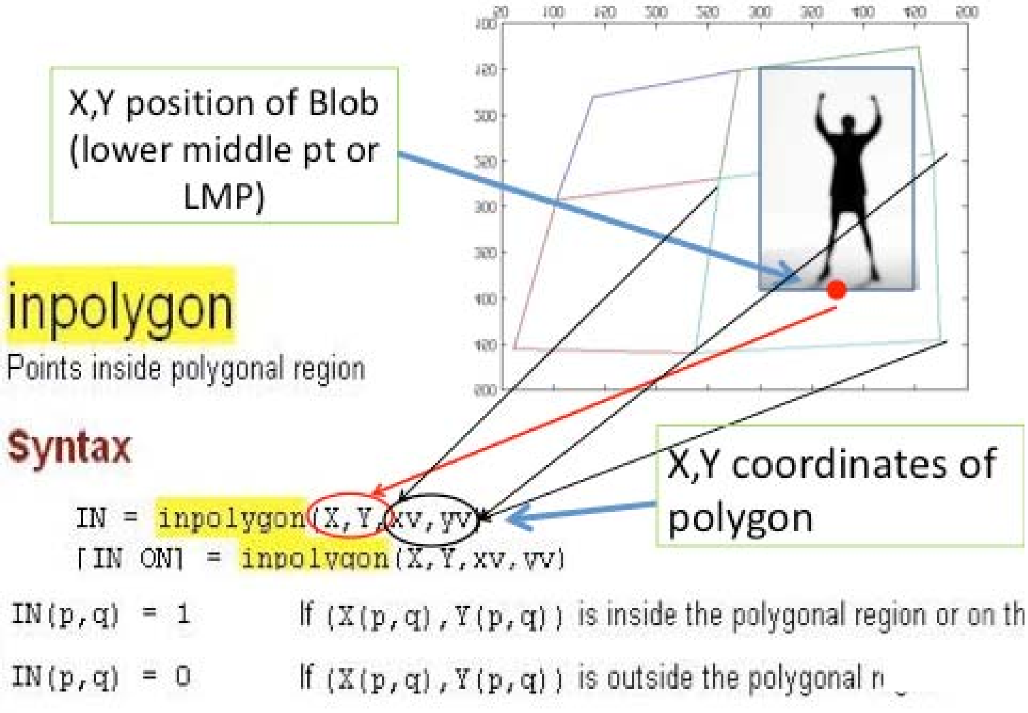
```
391.6969697   384.2865217   379.4695652   371.4457831   359.6666667   347.8291971   317.6608696   289.9578947   255.189759    429.0526316   431.96338
429.0526316   431.9633028   431.7571429   438.1851852   438.3939394   424.5526316   426.0917431   424.0095238   415.0511749   479.0526316   488.31764
431.9633028   431.7571429   438.1851852   438.3939394   424.5526316   426.0917431   424.0095238   415.0511749   483.3785124   488.3176471   484.14729
384.2865217   379.4695652   371.4457831   359.6666667   347.8291971   317.6608696   289.9578947   255.189759    214.0869396   431.9633028   431.75714
391.6969697   384.2865217   379.4695652   371.4457831   359.6666667   347.8291971   317.6608696   289.9578947   255.189759    429.0526316   431.96338
```

Y-Coordinates

```
52.39393939   67.39130435   83            102.8361446   128.0595238   162.2189781   203.7826087   263.0789474   340.4086024   58.75438596   64.155963
58.75438596   64.1559633    80.34285714   99.97777778   125.4545455   157.8684211   198.9357798   259.0582011   340.4543081   58.75438596   62.176470
64.1559633    80.34285714   99.97777778   125.4545455   157.8684211   198.9357798   259.0582011   340.4543081   463.6826446   62.17647059   78.248318
67.39130435   83            102.8361446   128.0595238   162.2189781   203.7826087   263.0789474   340.4086024   457.7816764   64.1559633    80.342857
52.39393939   67.39130435   83            102.8361446   128.0595238   162.2189781   203.7826087   263.0789474   340.4086024   58.75438596   64.155963
```

Figure 67. Camera Calibration Zone Cross-referencing

A main MATLAB function within the Blob Tracking program, called "GetZone" had to be created that uses the MATLAB "inpolygon" formula to evaluate if the lower middle position (LMP) of the blob (video track) is within any of the polygons of the specified camera's FOV.


X,Y position of Blob (lower middle pt or LMP)
X,Y coordinates of polygon

inpolygon

Points inside polygonal region

Syntax

```
IN = inpolygon(X,Y,xv,yv)
[IN ON] = inpolygon(X,Y,xv,yv)
```

```
IN(p,q) = 1        If (X(p,q),Y(p,q)) is inside the polygonal region or on th
IN(p,q) = 0        If (X(p,q),Y(p,q)) is outside the polygonal r
```

Figure 68. MATLAB "inpolygon" function depiction

The “GetZone” function returns, to the blob tracker, the *zone* where the LMP of the blob was located. The output is the designated number that represents one of the unique zones of all the zones in the camera’s FOV. Each camera has a unique FOV, and thus can have a unique number of zones.

This very labor-intensive process for mapping each camera FOV to obtain the calibrated zones was modified to include a routine that allowed the user to merely use the computer mouse to identify the zone boundaries for all the zones in a FOV. This is still, unfortunately, a manual process at this point, however, it only requires one step, which is to click, in order, the points which define each zone in a FOV. Once this procedure is complete, the matrices and associated files for the blob tracker program are automatically generated, which is a huge improvement in time savings and accuracy over the initial process.

Verification testing was conducted for the behavior analysis of targets with the new calibrated camera processed video data. Nine runs were conducted with three sets of three separate behavior cases, Normal, Abnormal, and Unknown. The system correctly classified the observed target behavior nine out of nine times, for 100% accuracy, exceeding the 90% detection threshold requirement for behavior analysis.

(3) Addition of Video and Audio collection "Kiosk." The goal of the Kiosk ECO was to extend the Watchman network, adding interactive video face recognition and audio recording and analysis capability. This ECO also clearly demonstrated the open architecture design of the Watchman, with the ease that an additional sensor and distributed processing suite could be integrated into the original system. This architecture is key to the WMSE POCS, showing how overall system functionality is enhanced, not degraded, by the integration of key capabilities into the system architecture.

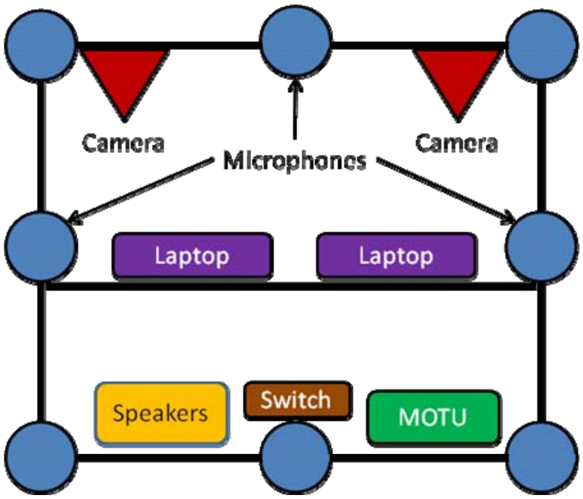


Figure 69. Video and Audio Kiosk layout

The audio/video kiosk consists of two cameras and eight microphones connected to two laptop computers, as depicted in Figure 69. The cameras and computers are connected by an Ethernet switch to the rest of the Watchman network, allowing the computers to share detection and analysis data with the Watchman server.

The cameras are Sony SNCRZ30N pan/tilt/zoom motorized cameras. They provide images to the video-controlling laptop using an Ethernet connection. They can be configured to send images at regular intervals or when they detect motion, and they can be controlled automatically or manually from the computer.

The microphones are connected to a MOTU audio mixer, which is then connected to the audio-controlling laptop via a FireWire connection. Software on the computer records audio information and displays a spectrogram during playback for analysis.

The video-controlling laptop uses timer objects in MATLAB to read and display the images sent by the cameras. Images are named with the time they were taken, and the MATLAB program searches for images with names matching the current time. Once a matching image is found, the MATLAB program runs the face detection algorithm. If the program finds a face, it commands the camera to zoom in on the face and provide more images. These new images are sent to the face recognition algorithm to record the subject's identity.

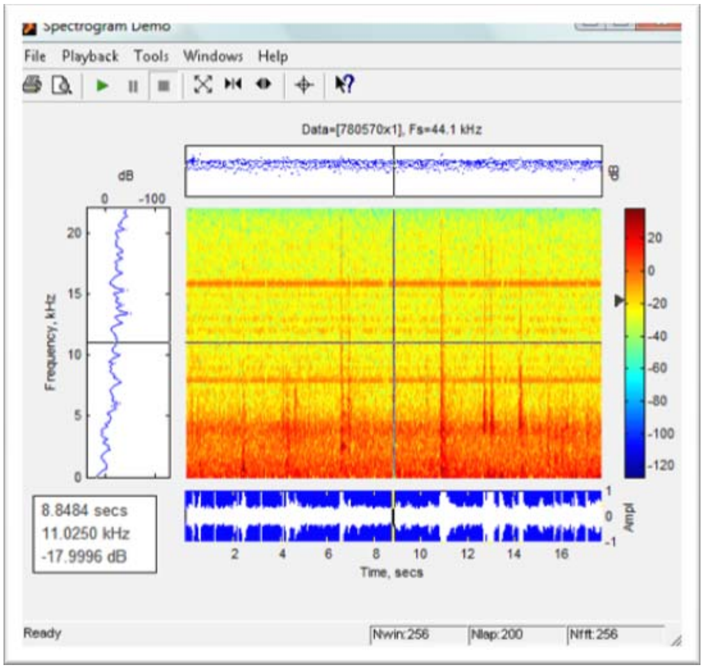
The MATLAB program that detects and recognizes faces also sends movement commands to the cameras to make them zoom in for a high-quality image. It uses Common Gateway Interface (CGI) commands sent using the Hypertext Transport Protocol (HTTP) commonly used to load web pages from the Internet.

The Audacity software program on the audio-controlling laptop controls the recording process and performs analysis of the recorded audio streams. These recordings can be saved an analyzed later to help verify the identity of the speaker. The Audacity software records from two of the eight microphones. Once an audio file is created, a MATLAB program converts the recording into a spectrogram showing the relative signal strength over the frequency spectrum as the audio stream changes over time. This spectrogram is illustrated in Figure 70.

Should they be desired for further project research, the existing kiosk provides the architecture for adding more advanced capabilities in the future. Video and audio capture and analysis can be improved and new functions added according to the needs of the user.

Identified potential upgrades include:

- Automatic mustering. Where the video-controlling laptop can perform its face recognition algorithm and then report its findings to the Watchman server for mustering purposes.
- Automatic audio recording, where the audio-controlling laptop can monitor background noise levels and capture audio streams only when it detects a significant rise in the audio signal.
- Audio source location. Where all eight microphones can compare the minute differences in the time of receipt of an audio signal to calculate the spatial position of the audio source. This position can then be translated into a vector for the cameras, which can pan toward the source and record video data.



Data=[780570x1], Fs=44.1 kHz

8.8484 secs
11.0250 kHz
-17.9996 dB

Nwin:256 Nlap:200 Nfft:256

Figure 70. Audio Spectrogram

(4) Blob Tracker upgrade to generate a color histogram. The Watchman blob tracker subsystem was upgraded to have the ability to generate a color histogram of detected targets ("blobs"). These changes allow more accurate blob differentiation and position analysis. More accurate blob differentiation will allow the system to be updated in the future to track multiple blobs simultaneously. The tracker accomplishes this by having the ability to recognize one blob from another by color differentiation. This capability, coupled with the more accurate position analysis will allow the watchman system behavior analysis subsystem to better classify a tracked blob's behavior.

The proposed functionality will create a color histogram by cropping a red, green, blue (RGB) color video still with rectangular coordinates. Those coordinates, [x y rows columns], will be provided by the existing blob tracker subsystem. The subsystem then performs a color histogram analysis of the cropped image using the MATLAB histogram function. The program will then provide a matrix at the network time interval summarizing this data and sending it to the Watchman server subsystem for eventual use by the BAM.

(5) Update WAES Server. This ECO incorporates the Watchman Server Microsoft SQL Server database and Microsoft Access frontend application updates required to support the other ongoing ECOs. This ECO included update to the current server data storage, processing and display. Specifically, modification of the database; tables and queries, forms and reports, and event code was required. The updates to the server database and application are listed below with their associated ECO.

- Microsoft SQL Server 2005 database
    - Detection table (tblDetection)
        - Camera Calibration (ECO 3)
            - Add field for zone
    - Color Histogram (ECO 7)
        - Add fields for histogram bins;
            - Eight (8) red channels
            - Eight (8) green channels

            - Eight (8) blue channels
- Field of View Optimization (ECO 1)
    - Add fields for x and y position of blob tracker
- Microsoft Access Project frontend application
    - Muster subform (frmMuster)
        - Camera Calibration (ECO 3)
            - Update to display zone for muster (i.e., 5.1.*A, where 5 is the node, 1 is the camera and A is the zone*)
    - Node Tracker subform (frmBullardSecondFloor)
        - Camera Calibration (ECO 3)
            - Update to display zone for tracking
            - Previously the greatest resolution the tracker could display was the camera’s entire field of view
    - Behavior Analysis Monitoring (BAM) subform (frmBAM)
        - Camera Calibration (ECO 3)
            - Add zone to Alerts and Recommendations received
            - ProcessQueries (qryNodeTrackerOne, qryNodeTrackerTwo, qryNodeTrackerThree)
            - Add zone to queries
        - Behavior Code (ECO 12)
            - Behavior Analysis Monitoring Export Reports
            - Add report to export zone to BAM for analysis (rptBAMzone) exported to “BAMzone.txt”

The existing interfaces, shown in Figure 71, between MATLAB, the SQL Server and the Access frontend application for writing to the database, as well as between the Access frontend application and the SQL Server for displaying data from the database were not modified. This demonstrates how significant upgrades to functionality within the system can be accomplished without the necessity to modify existing interfaces, which is a mark of robust design architecture.

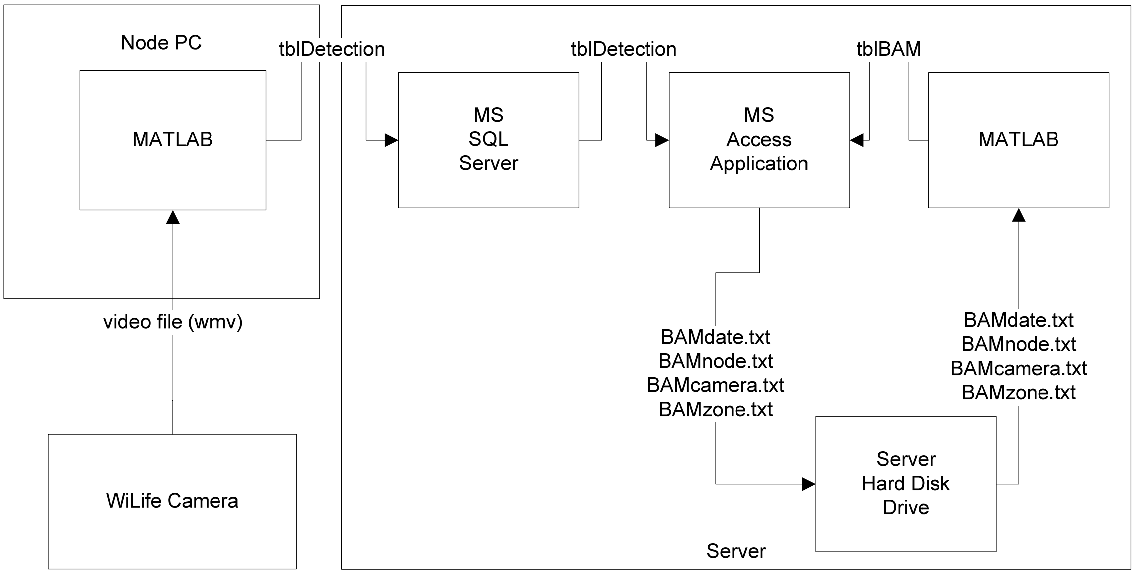


Figure 71. Watchman Server Interface Block Diagram

b. Integration of the Wireless Smart Sensor Network (WSSN)

This follow-on Systems Integration upgrade project focused on three primary enhancements to the current Watchman Maritime Smart Environment (WMSE) concept development and demonstration system. The WMSE system was meant to demonstrate the capabilities required to meet a target behavior analysis problem on a small, measurable scale as a proof of concept that could reasonably be expanded and developed into a full-scale prototype development system.

These enhancements are within the *Analyze/Classify*, *Provide Indication and Warning* (I&W), and *Respond* functions of the system architecture. A primary purpose of this project is the addition of the Wireless Smart Sensor Network (WSSN) to include upgrades to the Behavior Analysis Module (BAM) analysis and integration with the I&W system.

Integration of the WSSN and the upgraded BAM to the current Watchman System was accomplished using a systematic, modular approach by which the separate system functions were developed and implemented individually. Integration was accomplished after initial functions and interfaces were in place and tested. A summary of the required enhancements that were achieved follows:

1. Analyze/Classify
   - Behavior Analysis Module (BAM)
     - Update BAM software to more accurately analyze targets
2. Provide I&W
   - Expand Command and Control (C^2) network to disseminate I&W data and issue tasking
     - Update Watchman Application to disseminate I&W data and issue tasking to the WSSN
3. Respond
   - Develop a WSSN of response agents to collect additional data as required on a Contact of Interest (COI), as directed by the Watchman System
     - Integrate WSSN response agents to react to identified COI as required

Minimizing the scope of this project maintained the project within the Network-Centric Systems Engineering (NCSE) Lab in Bullard Hall, room 201L. Figure 72 shows the layout of the NCSE Lab and positions of the cameras. For this project, Camera 6.1 and Camera 6.6 were utilized to capture the video of the COI for processing at the Node C^2 computer.

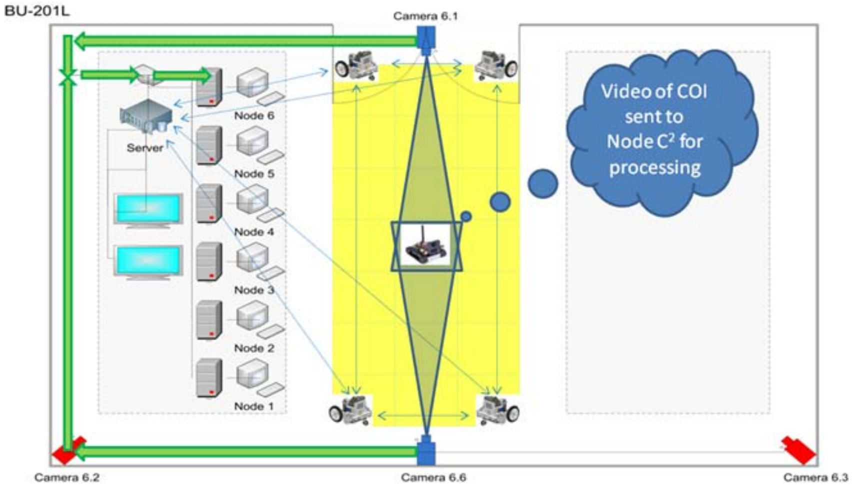


Figure 72. COI video processing

To recapitulate basic system operation: video from the Node cameras is sent to the Node C^2 computer where the COI detection and tracking takes place. The intelligence is extracted and written to the Watchman Server Database. The Watchman Application queries the data, creates the BAM utilization files and exports them to the BAM Classifier. The BAM Classifier analyzes the behavior strings from the BAM files and writes the classification to the Watchman Server Database. The Watchman Application then provides analysis for COI classification (Normal, Abnormal, Unknown) to operator for action.

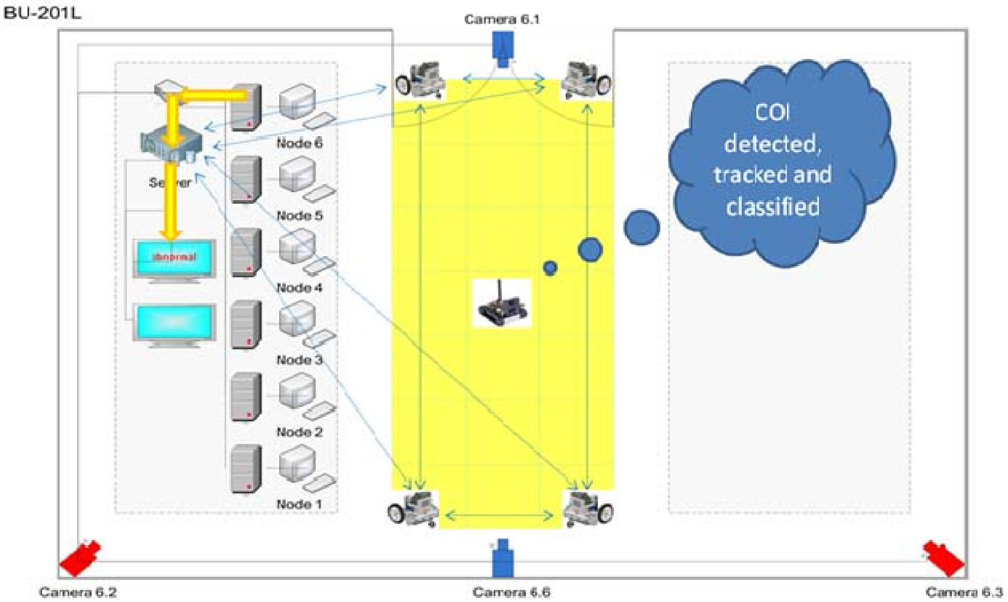


Figure 73. COI detection, tracking and classification

As part of the Respond function addition to the Watchman for this project, the Watchman Application sends a Tasking Message with the location of the target (zone) and task (data requirement) to the agents within the WSSN. This message is sent automatically for an "Abnormal" classification. For an "Unknown" classification, the system will ask for operator initiation (at their discretion) of the Tasking Message.

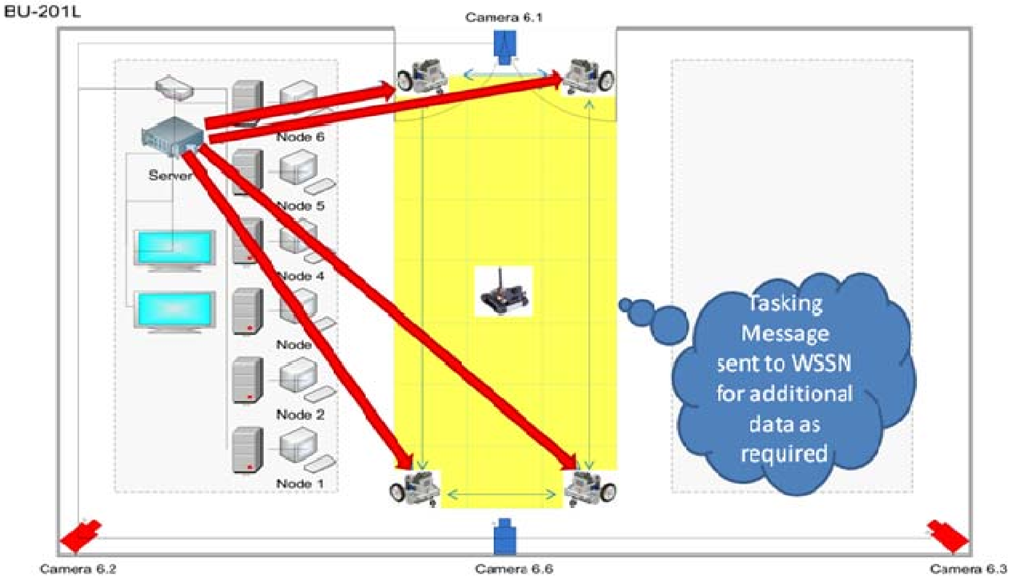


Figure 74. Tasking Message sent to WSSN

The WSSN consists of agents of two configurations for accomplishment of different tasks. The agents within the network determine which agent will accomplish the task required from the Tasking Message. The agent maneuvers to the COI, performs the required task and sends the data back to the server for processing.

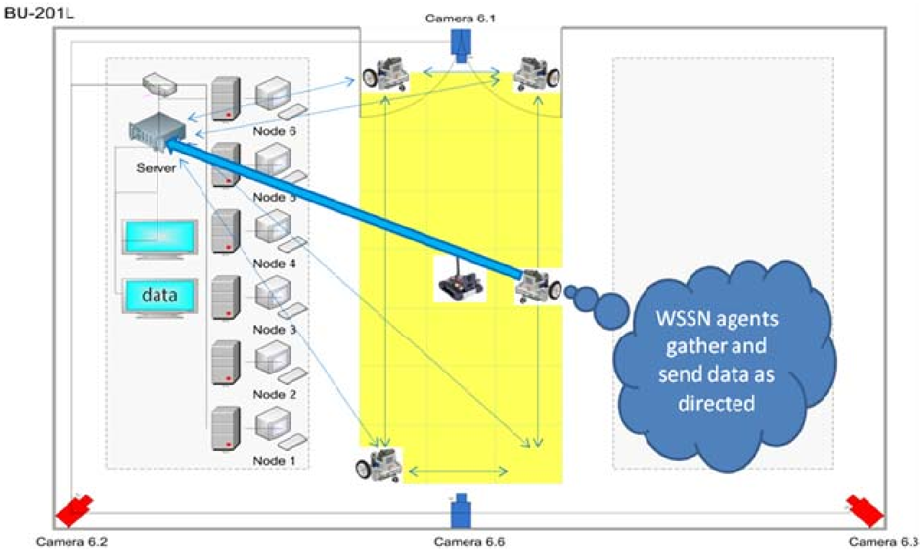


Figure 75. WSSN agent responds to Tasking Message

An overview of the current Watchman System and integration requirements for the project are described in Figure 76, the Watchman Integration Upgrade System Overview. The red and yellow arrows illustrate the two primary integration points for the system. The first integration requirement for the system, indicated by the red arrow, was the automated non-real time import of the video files from the fixed WiLife cameras into MATLAB for processing. This requirement necessitated design and implementation of new software to accomplish the integration, however this function was later determined to not be required for WSSN integration, and was therefore deferred to a later upgrade project. No hardware requirements were identified during the analysis as required for this deferred integration requirement. The yellow arrows and text indicate the second integration requirement, which was implemented. First, the MATLAB BAM Classifier had to be updated to support integration of Node 6 zone mapping for Behavior Classification. Next, the initiation of the WSSN Tasking Order from the DMS Application to the WSSN C^2 Node and transmission to the WSSN Agents had to be automated. Finally, the WSSN Agent reply to the Tasking Order also required automation to support full integration. This second integration requirement was accomplished by new and modified software development with only the minimal hardware requirement of an installed COTS Bluetooth communication Universal Serial Bus (USB) device.

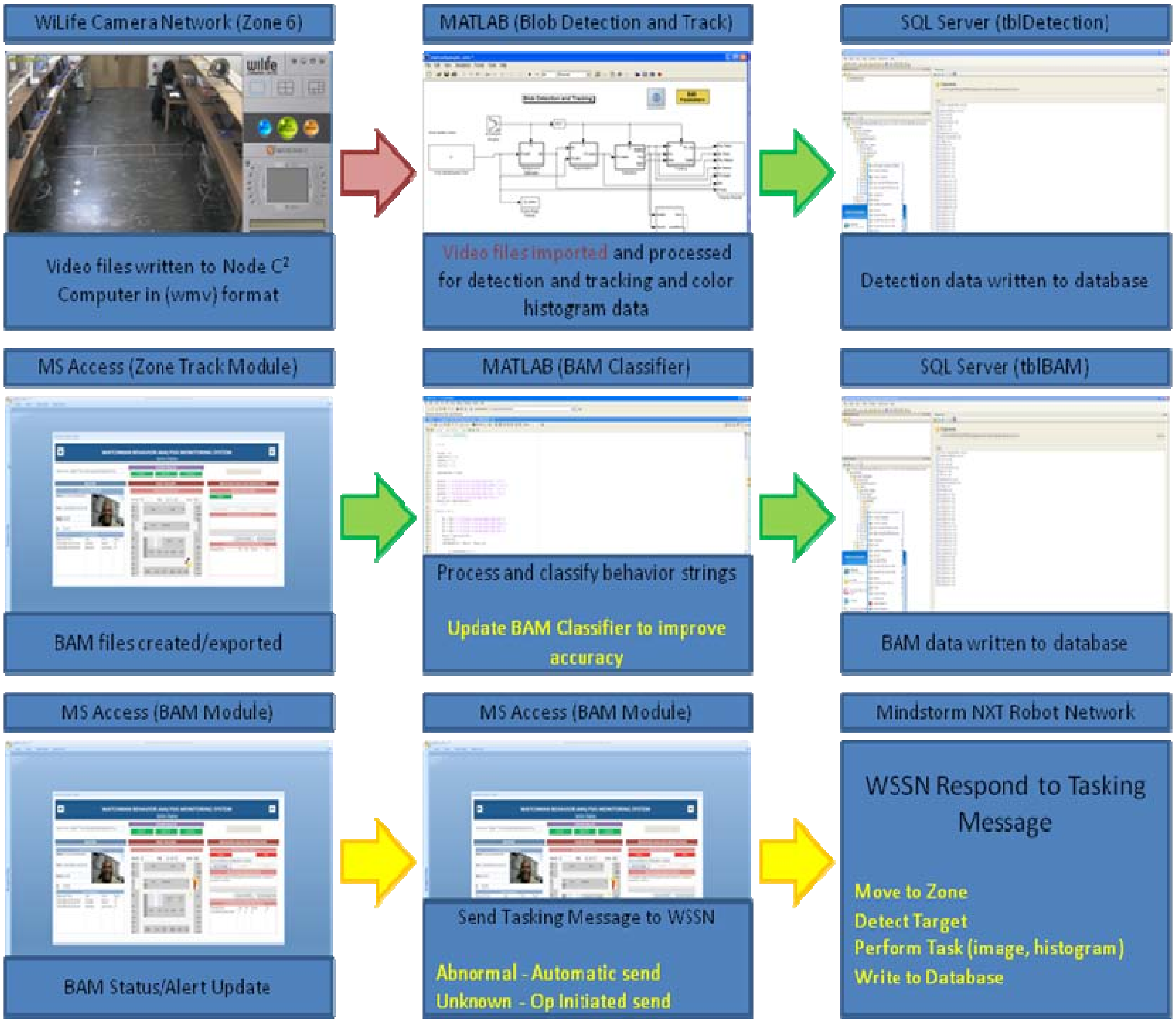

WiLife Camera Network (Zone 6)
Video files written to Node C² Computer in (wmv) format
MATLAB (Blob Detection and Track)
Video files imported and processed for detection and tracking and color histogram data
SQL Server (tblDetection)
Detection data written to database
MS Access (Zone Track Module)
BAM files created/exported
MATLAB (BAM Classifier)
Process and classify behavior strings
Update BAM Classifier to improve accuracy
SQL Server (tblBAM)
BAM data written to database
MS Access (BAM Module)
BAM Status/Alert Update
MS Access (BAM Module)
Send Tasking Message to WSSN
Abnormal - Automatic send
Unknown - Op Initiated send
Mindstorm NXT Robot Network
WSSN Respond to Tasking Message
Move to Zone
Detect Target
Perform Task (image, histogram)
Write to Database

Figure 76. Watchman Integration Upgrade System Overview

A system demonstration test was conducted to validate the functionality of the WSSN subsystem. Four runs were conducted under the criteria for Alert-Type Task Initiation (Abnormal alert or Unknown alert) and Data Collection Task Type (Image or Histogram). The system conducted all four runs successfully with the agents correctly receiving each tasking order based upon alert message type; correctly determining which agent would respond through autonomous communication and decision making; correctly navigating to the target; and correctly collecting and transmitting the requested target data from the tasking message.

This section discussed the WMSE POCS in order to provide the network context as the basis of the ABA integration research effort, which is the focus of this thesis. The system was designed, developed and tested as a proof of concept, and all systems and subsystems were tested and validated according to the determined requirements under controlled laboratory conditions. The next section discusses the implementation of the WMSE POCS concept in a notional operational environment, analyzing the possible alternative approaches that could be taken in that implementation. This implementation discussion will expound upon the MBAM portion of the system and how it can integrate into an MDA network as the specific area of concentration for the thesis research concept.

B. OPERATIONAL IMPLEMENTATION

Although the WMSE POCS was successfully evaluated, exceeding all performance metrics in a laboratory environment as established in the previous section, the crux of this thesis is not necessarily the success or failure of the WMSE POCS itself, from a standpoint of some new, cutting edge capability that can be brought to the fight of MDA. While the WMSE POCS, in itself, does represent some move forward in the field of automated behavior analysis and processing, the real issue for stakeholders is how this type of capability, whether it is a future iteration or increment of the WMSE system or one of the other behavior analysis capabilities discussed earlier, such as PANDA or CMA, may be actually fielded to meet the requirements of the users that need them. This challenge is uniquely a Systems Engineering one, and one which this thesis works to address, in understanding and analyzing the process of transitioning from user needs to requirements analysis, right on through architecting, design, build, and integration, and finally verification and validation of the systems solution to meet the original specified user need. Using the SE Process lessons learned for the WMSE POCS, which were outlined in Chapter III, this section explores the most feasible means of bringing a MDA enhancing capability, via a MSE, from concept to operation by properly applying sound SE principles.

The two main SE methods by which a MSE could be implemented for operational use are either a complete, end-to-end, “from the ground-up” system development, or a smaller scale integration of solely the Maritime Behavior Analysis Module (MBAM) into the current Maritime Network architecture that accomplishes MDA. The first method would encompass the original development of every aspect of the entire MSE. This would include new or upgraded sensors of all types, new or upgraded data fusion, behavior analysis, reporting, and reaction capabilities, along with all of the requisite network communications, command, and control (C^3), and the very latest in distributed data processing that would make a cutting-edge MSE network function. The second method would only require the single MBAM (a “new” component) to be added and integrated into the current Maritime Network, under the FORCENet construct. However, as simple as this may sound, in order for this integration to be successful, the MBAM would have to be compatible with all of the legacy interfaces of the network that will have inputs to and receive outputs from the MBAM, without adversely impacting the other critical functions of the network.

The next two sections look at these two implementation methods from a SE perspective and discusses the pros and cons of each, making recommendations for which circumstances each method would be preferred.

1. Complete “From the Ground-Up” System Development

This method has several advantages from an SE point of view; the most obvious being that this method allows for all elements, components, and subsystems to be developed, designed, and integrated as a complete system, from “cradle to grave.” This method, when time and budget allow (which unfortunately is often not the case) is the most preferable for Systems Engineers. This is due to the fact that all systems elements, such as interfaces, hardware components, software languages and modules, etcetera can be architected, designed, built, and tested in a simultaneous, coherent, and coordinated manner. This simultaneous SE development across the whole of the system leads to a greater understanding and knowledge by Systems Engineers of the system, as well as greater access to and control over the many systems' aspects, such as hardware and software functional and interface design, system-wide test planning and execution, system-wide requirements definition, and many others. Having a thorough knowledge of, and greater access to and control over all various systems components helps to prevent many of the mistakes that occur throughout the SE process, such as misunderstood requirements, missing or incomplete architecture linkages, poor design direction, incomplete or inadequate interfaces, inadequate or ineffective testing, and much more.

Another advantage found with this type of total system project is that the effort is at least managed, if not executed, by the same organization, such that the same individuals in the project office are working and coordinating on all aspects of the system, providing continuity throughout the entire SE development process. Even though these types of very large, comprehensive projects can span many years through the development cycle, which means turnover of engineers, project managers, and others, very rarely is there wholesale turnover of all the personnel in an office or organization. This means that there always exists some "corporate knowledge" of the technical aspects of the project throughout the development lifecycle process, which greatly reduces the chance of error in Systems Engineers "relearning" the same mistakes over and over again.

Quite possibly the principal advantage to engineering an entire AI system to create an end-to-end MSE would be the ability to architect, design, and build the infrastructure upon which the remainder of the AI systems solution could be built, a' la the AI Systems Solution Pyramid model discussed in Chapter III. This total systems solution model affords a customized infrastructure whereby the intelligence automation application subsystems (i.e. DIPR) could be superimposed and would ideally function much more flawlessly, not requiring additional (and quite possibly complex) interface hardware and software. In this model, given the availability of all required concurrent technologies (a considerable assumption), the detection sensors, identification (symbol production) processing, prediction algorithms, and reaction systems all could run on the backbone of an AI systems infrastructure tailored for their unique operation.

The main disadvantage from this approach is the inordinate and often prohibitive cost in time and money that such a large undertaking incurs. These very large and comprehensive projects are so vast in scope that initial cost estimates cause budgetary “sticker shock” for stakeholders, making these worthwhile projects very difficult fiscally to garner enough budget support to get off the ground. Additionally, project stakeholders also want needed capability in the hands of users quickly, which makes this type of full-scope approach unattractive. The time that such a development effort takes, in order to fully build all the architectures, design all of the components and interfaces, then fully test and validate, with ample additional amounts of time for fixes and re-testing, frustrate the efforts of the project to maintain the favor of stakeholders with competing priorities.

This disadvantage can sometimes be overcome by using SE development strategies such as “evolutionary acquisition” or “spiral development” or any of the other terms, which happen to be in vogue at the time for incremental development.[95] If the total system scope can be broken up into phases, with different levels of capability developed and delivered to the user over time, then the comprehensive system approach has a much better chance of showing a more near-term return on investment for the stakeholders, and therefore offering a much better chance of programmatic success, regardless of the technical advantages.

Despite the technical synergistic advantages of the complete systems approach, certainly there are drawbacks as well, considering the complexity of such a large endeavor. As with any system of systems (SoS), the more systems involved, the more complex the overall SoS will be.[96] Given the inherent complexity of the types of individual systems within this type of SoS, and the coupling and cohesion that must occur with any networked SoS, the combined complexity of a comprehensive MSE SoS would be very complex, indeed. It is this complexity that leads to the long systems development

[95] INCOSE Systems Engineering Handbook, version 3, June 2006, Edited by: Cecilia Haskins

[96] Ibid.

cycle and extraordinary cost, which was mentioned earlier. System stakeholders would be required to conduct an analysis of the trade-offs between complexity, schedule, and cost, as compared with the advantages of control, coherency, and coordination and of a total system development concept before committing to that path.

An example of this type of approach may very well be the CMA JCTD that was discussed in Chapter II. This system is a comprehensive MDA enhancement system, which was tested and intended for fielding as a total network solution, complete with functionality for autonomously aiding operators in determining intent, much like the MBAM of the WMSE. It is unknown how long this system will take to fully develop and implement, or if an acquisition strategy has been adequately developed with proper SE input and oversight to come up with an effective evolutionary capability-phased approach to development that will allow this promising suite of capabilities to get into the hands of MDA practitioners in a reasonable time frame in an environment of constricting resources. Nonetheless, it would not be recommended to apply this type of approach to the WMSE POCS, or any other MDA enhancement system, until the technology for all elements of the architecture are mature enough to support system-wide development and implementation.

2. Integration of MBAM Into Current Maritime Networks

The other method of implementation that will be explored is the integration of a matured Maritime Behavior Analysis Module (MBAM) into a local tactical MDA network, such as was described in the DRM OV-1 diagram, to create a MSE. This method would utilize all of the current and future expanding MDA capabilities, within existing net-centric architecture, to enhance the MDA mission with the addition of an ABA capability. First discussed are the pros and cons of this approach, and then an example is offered of how this might be accomplished with the WMSE POCS architecture as a model of a notional operational implementation.

The main advantage of this method is the minimization of schedule and cost by utilizing existing MDA sensors, data processing and management networks, C^2 structures, etcetera. This cost minimization is not only found by avoiding additional

R&D costs of new MDA network assets, but also in the deployment, training, sustainment, and all of the other life-cycle costs associated with the deployment of new systems. Of course older systems become obsolete and can therefore become more expensive to sustain; however, the proper integration of an MBAM capability should employ open-architecture principles[97] such that older components of the overall MDA network can be removed, replaced, and/or upgraded with minimal to no impact to the MBAM interfaces and associated capability.

The schedule minimization advantage can be found in the fact that the MBAM integration can be a stand-alone effort, without any impact to the current systems or their operation. By understanding the type and format of input that the MBAM requires, as well as the output it delivers, “bridge” code could be developed to translate current sensors fused track data into the format that could be processed by the MBAM. Likewise, additional bridge code would be developed to take the MBAM output and provide it, in the form of alerts, to the current C^2 systems’ GUIs for operator action. All that would be required would be the interface data exchange information (which should not have a proprietary access restriction, due to “open architecture” acquisition rules)[98] so that software development engineers have accurate insight into the data fusion outputs and the C^2 GUI inputs. Once the input and output bridge code was tested and validated with the MBAM functionality, the system could be installed and deployed.

A consideration that should not be taken lightly is the challenge of customizing the MBAM for use in its particular tactical environment. Recalling from the AI Systems Solution Pyramid presented in Chapter III, the top level represents the customization of applications that are required for their specific implementation. Certainly this would be true for any MBAM application, as the behaviors of potential threats in the Straits of Hormuz, for example, would be quite different than those of the Caribbean, as well as those off the coast of Somalia. Part of the entire implementation process would be the creation of databases for both normal, abnormal, and known threat behavior for any

[97] Naval Open Architecture Contract Guidebook, version 1.1, October 25, 2007, PEO-IWS 7.

[98] PEO IWS, Naval Open Architecture Contract Guidebook.

potential theater MDA network into which the MBAM would be integrated. These databases would also need to include customized cost calculations correlated to the unique behaviors and environments in which those behaviors would be found. While this challenge is by no means a simple one, it is the same challenge that would be found and necessary to overcome with a "ground-up" full system development implementation. The ability to customize the MBAM as a stand-alone application that can then integrate into existing networks is still a significant advantage for cost and schedule reduction.

Another advantage of MBAM integration is a technical advantage, which was alluded to previously. The MBAM would bring a significant capability to enhancing MDA, however, it would not necessitate the excessive alteration of any existing interfacing systems. To the contrary, the only anticipated change to existing infrastructure would be the space allowance for the system, data connections for network connectivity with the system, and upgrades to the C^2 system to allow for display of MBAM output alert data. Additionally, the physical locations (e.g., databases) of features preexisting would need to be known in order for the MBAM to access them.

Given the state of technology today, as well as emerging technologies, it is recognized that there are varying degrees of automation that may be found in the systems present in any given MDA network, when attempting to integrate ABA capability. This variability will lead to different cases that must be dealt with in the implementation.

Figure 77 depicts the integration of a MBAM under the DIPR construct, utilizing the bridge code under differing technical circumstances, as previously described. The legend in the diagram shows those software elements that would be required to implement bridge code in order for the MBAM to function within the existing MDA network. As the diagram shows, all of the interfaces with existing or "legacy" systems would require bridge code in every case, for MBAM functionality without legacy system or network disruption. In addition to interfaces, bridge code would be required to accomplish the *Identify* function of DIPR in every case, which is, in effect, the translation of fused sensor data into intelligent states (symbols), as discussed in Chapter III. This implementation of *Identify* for symbol creation would be the translation of fused sensor data into MBAM-usable format, described in the previous paragraph.

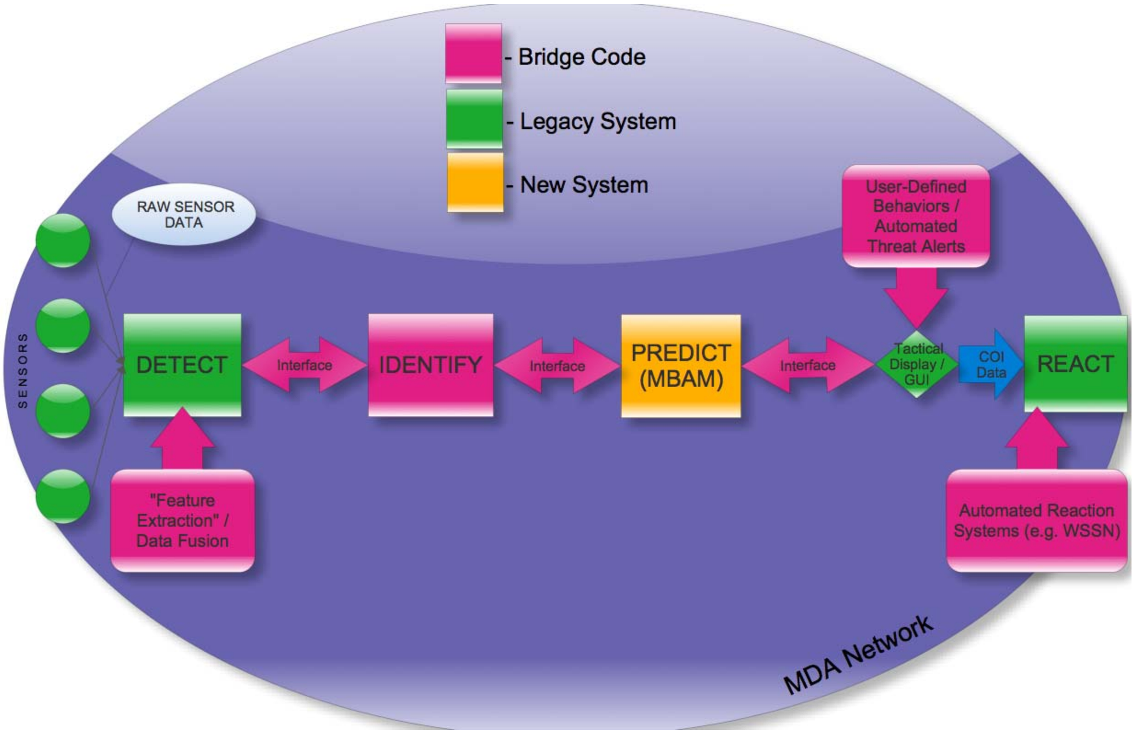


Figure 77. Integration of a MBAM Through the Use of "Bridge Code"

There are some cases, however in which bridge code may or not be required for the legacy systems themselves in order for the MBAM to function properly and effectively. These cases are depicted in the diagram by the rounded rectangles with arrows directed to the systems that would be affected. For example, older legacy systems that perform the *Detect* function may not have the capability to process raw sensor data for feature extraction or they may not have the capability to fuse those features. Bridge code could be applied "upstream" of the *Identify* function that would process the sensor data to extract the features required and/or fuse those features to provide fused contact data in the format that then could be processed by the *Identify* function for further transmission to the MBAM. If the *Detect* system(s) have this automated feature extraction and fusion capability already, this step of integration would not be required.

In other cases, stakeholders may desire for the *React* subsystems to have more automated capability, or for the user interface(s) to have some automated alerting capability or features to allow the user to customize the defined behaviors. In these cases, bridge code could be applied to the GUI or *React* subsystem, as shown in Figure

77, where integration of tailored tactical display interface functionality, or integration of an automated reaction subsystem, such as the WSSN described earlier in the chapter, could occur.

Given the fact that the MBAM is essentially a processing capability, physically integrating it would entail installing a platform of sufficient memory and processing power to hold and process the data for the hundreds, if not thousands of eventual maritime behaviors it would need to analyze, along with the requisite data connections. Display data could be accomplished via two means. One way would be via a stand-alone monitor system, which would be easiest and quickest to install and integrate, yet would require additional training and monitoring, and is therefore not recommended. The preferred method would be to provide output data information to the display software vendor via the output bridge code and work with that vendor to incorporate MBAM output data through the bridge code as part of the current display format (see Figure 77).

A disadvantage that arises from integration into an existing network is the danger of data corruption from the introduction of a non-compatible system and therefore performance degradation of the existing systems. For example, if the preferred method of MBAM output data integration were to change the software of the C^2 displays to accommodate MBAM alert data, this change to would require ongoing software maintenance of the C^2 software as it pertained to changes in the MBAM software that affected output to the C^2. This additional functionality would potentially increase the chances of software failure due to the incorporation and maintenance processes. Though this would increase the maintenance burden and the risk of failure to a degree, incorporating this small additional maintenance task would still be far outweighed by the benefit of seamlessly integrating MBAM data into the C^2 picture for MDA. However, this risk is avoided altogether through the use of bridge code to translate the output data into that standard display input format. Other data corruption risks are minimized, if not eliminated, by the fact that interfacing with the MBAM does not directly influence the existing systems. Of course, full risk mitigation of network corruption could be accomplished through the implementation of some type of hardware or software "bypass," whereby operators could effectively remove the MBAM from the network at any time, should data corruption or functional degradation as a result of MBAM be suspected. Nonetheless, through the use of bridge code, the existing systems maintain their data and functional integrity, without being required to alter their interfaces or data processing structures.

Consider now how this might be accomplished with the WMSE POCS architecture as a model of a notional operational implementation, referring to the diagram of the functional architecture for “Enhance Domain Awareness,” Figure 77. The “Classify” function is expanded to show its required sub-functions. The “Detect,” “Track,” “Respond,” and “Perform Overhead Functions” main functions represent those functions that would be performed in a current, real-world, MDA network. The Detect function would be accomplished via any of the various sensors inked into the network, such as radar on an MPA aircraft, FLIR from a rotary-wing UAV, ESM from a surface combatant vessel, etc. The Track function could be performed by any of the platforms mentioned with their onboard processing capability, as well as through sharing the information via data links, and fusing the data into a COP. The Respond function could also be accomplished by any of the platforms already mentioned, as well as other platforms or assets, air, surface, or subsurface, within this particular network. The Perform Overhead Functions function would be the shared responsibility of all assets and platforms within the system to maintain their connectivity and data quality within the network.

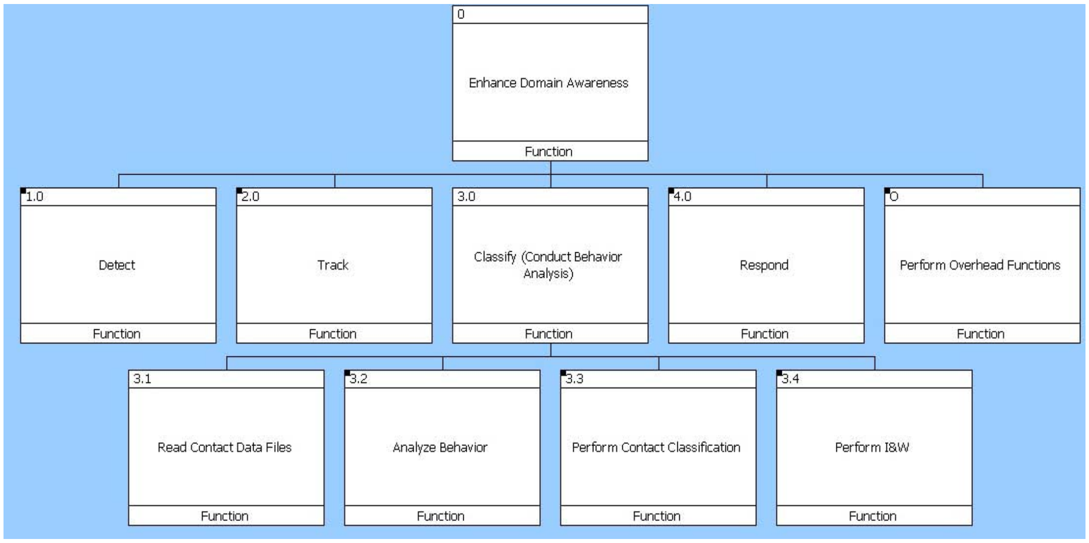


Figure 78. Enhance Domain Awareness, Classify Expanded

It becomes evident now, how the MBAM function fits within this architecture, performing the "Classify" function. There are certainly many means of classification, such as friendly, neutral, surface, air, commercial shipping, fishing vessel, etc.; however, for this architecture discussion, "Classify (Conduct Behavior Analysis)" will be limited to just what the label implies, conducting behavior analysis for the purpose of classifying a contact of interest (COI) as a threat, potential threat, or non-threat. As we look at the sub-functions of Classify that need to be performed, we can see that those sub-functions would be accomplished as described previously, by the MBAM and the required bridge code to interface with the other functions. "Read Contact Data Files" would be a bridge code function, taking the track data from the Track function and converting it into a database format that the MBAM could then pull from and execute the "Analyze Behavior" and "Perform Contact Classification" sub-functions. Then the "Perform I & W" sub-function would be yet another bridge code function that would take the MBAM classification data, convert it to a format that the C^2 display system could process, and then output it to the C^2 system for the "Respond" function to react.

This notional operational implementation is identical to the NCSE Laboratory implementation, where COTS cameras provided pre-formatted video data, and bridge code was created to take that video data and convert it to a format (i.e., *Detect* and

Identify of DIPR) that the BAM could process, with no degradation or change in the cameras' base functionality. Likewise, the BAM output (i.e., *Predict* of DIPR) was converted, via additional bridge code, to a format that the existing C^2 system could process and display for operators, and then automated robotic sensors, to react (i.e., *React* of DIPR). The proof of this concept provides a promising Systems Engineering path for bringing much needed MDA JIC capability to the fleet in a very effective manner.

This chapter presented the elements and development of the WMSE POCS and an analysis of potential operational implementation. The next and final chapter summarizes the information presented in this thesis, as well as provides conclusions and recommends further areas of study in the vein of this research.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. SUMMARY AND CONCLUSIONS

Chapter VI summarizes this work by reinforcing the need for a Maritime Smart Environment (MSE) system through integration of Automated Behavior Analysis (ABA) capability into existing Maritime Domain Awareness (MDA) networks to enhance MDA. The chapter reiterates how this approach will fulfill certain capability gaps, and the importance of developing the technology to scale and refine the Proof of Concept System (POCS), and the Maritime Behavior Analysis Module (MBAM) in particular, for operational use. The chapter also identifies and describes areas with potential for further research, development, and testing.

A. SUMMARY

This thesis described the research associated with the Watchman Maritime Smart Environment (WMSE) POCS project, along with the SE processes used to implement and demonstrate it, in order to establish the feasibility of integrating an ABA capability (akin to the MBAM) into an existing Net-Centric environment within the Maritime Domain. The integration of this capability, with its demonstrated potential to recognize particular behaviors and alert operators to them, reveals the opportunity to further develop this technology and systems integration concept to fill sorely needed MDA Joint Integrating Concept (JIC) capability gaps of aggregating, displaying, and analyzing maritime information in order to understand the maritime environment and identify threats (MDA-003C) and predicting activity within the maritime domain (MDA-004C). Meeting the numerous MDA mission requirements that these gaps represent will most assuredly enhance Maritime Domain Awareness by improving situational awareness, better utilizing the capability of networked sensors and the multitude of data they provide, as well as making operators more efficient in their duties to respond to real threats, rather than ineffectively attempting to manually monitor and react to threats arising from the whole of the maritime domain.

In brief summation: The Systems Engineering (SE) process was executed, iteratively, beginning with the user needs from the MDA JIC capability gaps, leading to the required operational activities to fill those gaps, which then derived and then mapped to the identified particular standard Navy tasks required. These tasks were then mapped to the functions and subfunctions, creating a functional architecture, that linked to subsystem processes and requirements, These required processes and system performances requirements led to analysis of options for components in several areas of the system. Ultimately this selection, combined with the constraints of pre-determined subsystems and system components (e.g. COTS hardware and software) the physical architecture was implemented in the design, build, integration, and eventually test and validation of the POCS.

After a brief introduction explaining the operational concept and mission need germane to the thesis, the development of the case began with an explanation of the current MDA capabilities and practice with a survey of the various representative platforms and sensor systems used to accomplish MDA in the different media - space, air, land (coastal), surface, and subsurface—within the maritime environment. This survey also took a brief look at some of the current Research and Development (R&D) initiatives for MDA enhancement that incorporate or attempt to utilize some form of automation for sensor data processing and fusion, as well as feature extraction and pattern or behavior recognition and analysis. These initiatives are at differing stages of development, and have each demonstrated varying levels of success in their implementation. The common thread with all of these systems, however, is that they utilize a "whole system" development approach, rather than attempting to integrate a single, specific capability, such as data fusion, or behavior analysis, into the existing MDA net-centric environment. This distinction was key to explain the unique SE approach for ABA that this thesis presents, as distinguished from other ABA-type initiatives. Additionally, although these systems employ a greater degree of automation than with previous systems, they still lack the complete, end-to-end automated intelligence system (from sensors all the way through to reaction systems) represented in the *Detect, Identify, Predict, React* (DIPR) model discussed in this thesis. Even though the approach recommended here is integrating ABA (*Predict*) capability into what would typically be a primarily non-automated network, any introduction of or increases in automation in the other subsystem areas (*Detect, Identify, React)* is highly desirable and would greatly increase the effectiveness of the ABA capability. For example, the more object (e.g., small vessel contact) features that are automatically extracted, the less manpower required to detect, identify, extract, and process the required features to support ABA.

It was then vitally important to discuss the SE approach in the conduct of the actual research, as well as how this approach, consisting of a concept and process, could be used as a model for physical implementation of real-world mission-capable ABA. This explanation was accomplished by first discussing the SE of Artificial Intelligence (AI) and ABA systems in general, exploring a notional automated net-centric system under the AI Systems Engineering 'pyramid' construct, and its fundamental component, the DIPR AI software approach model. From this foundation, the application of these concepts in a SE process model for the specific WMSE POCS was elucidated, presenting the type of model used, then describing each step as it pertained to the POCS, as well as process deviations, improvements, and lessons learned.

The next chapter expounded upon a key part of the SE process by describing the development of the WMSE POCS system architecture. This architecture description showed the functional, process, and operational decompositions that led to the design and build of the initial Watchman system. The description further explained how subsequent iterations of the requirements and functional allocations for system upgrades produced refined software architecture, as well as an integration architecture. This architecture, as well as its process of development is a vitally important element of ABA operational implementation. The specifics of this architecture are outlined in the very detailed System Description Document (SDD).

The next chapter thoroughly described the entire Watchman POCS, from initial concept to latter upgrades, additional capabilities, and improvements, which demonstrated its proven functionality in analyzing and recognizing behaviors in a network-centric engineered laboratory environment. The chapter documented the test results from both the initial system, as well as system upgrade testing. Test results of 90% facial recognition success, 100% behavior analysis success (both with initial and upgraded MBAM with enhanced spatial resolution), and 100% Wireless Smart Sensor

Network (WSSN) response success, confirmed the system, in all its design iterations, exceeding determined performance metrics. Delineating all of the components and aspects of the system was necessary to provide the evidence for the SE approach concept of integrating and ABA capability into a Net-Centric Warfare (NCW) milieu. The latter portion of the chapter addressed operational implementation of the POCS, by describing the two types of operational implementation System Engineers could expect to encounter when attempting to implement an ABA capability. These alternatives were analyzed, with the resultant recommendation of the singular integration of an ABA capability into the existing NCW environment, as the most cost-effective and mission-capable implementation solution.

B. CONCLUSIONS

The task of monitoring the maritime domain and remaining vigilant to the threats to our national security explanation that emanate from it will remain a vital mission area for any foreseeable future, exemplified by the description of the MDA JIC capability gaps and how they were addressed in this thesis, summarized in the previous section. The maritime domain is an ever-increasing net-centric environment, with a continually growing number of sensors, operating nodes, communications links, providers, and consumers of information. With these growing numbers of both fixed and mobile capabilities, there will continue a deficit in humans, bandwidth, power, and intelligent centers to deal with this cascade of data.[99] There can be no doubt that MDA is a very challenging field, with the majority of the earth’s surface falling under its purview. Highlighted in this research are the many advances that have been made in the field, in the way of more highly capable, longer range, and extended duration sensors; more robust and reliable communications; higher bandwidth and faster networks with improved data sharing; and lastly much higher fidelity and accuracy in the fusion of the sensor data which is passed among these networks. With all these advances, the data available is just too voluminous for humans alone to process and react to manually,

[99] R. Goshorn, “Smart Robot Workshop,” Introduction Brief, Naval Postgraduate School, April 2010.

sifting the “wheat from the chaff,” and be expected to accomplish effective operational decision making regarding maritime threats to national security and international peace and trade.

The ability, through scaled development, of integrating a DIPR-type ABA capability would certainly enhance the application of MDA through fulfillment of stated JIC capability gaps. This integration methodology, put into operational practice, would allow a more seamless implementation of maritime prediction capability, less disruption of current sensor data collection capability, processes, and operations, and ultimately a satisfaction of the critical mission needs that exist in maritime situational awareness, threat recognition and response.

C. AREAS FOR FURTHER RESEARCH

Throughout the course of this thesis research, several other areas related to, but outside the scope of this project arose that merit further study and exposition. A few of the most promising and pertinent topics are offered here for the reader to pursue, as desired.

The first important area for future work is the enabling of a “learning” function within the Watchman BAM. A starting point for developing this capability would be the discussion on “learning” from Goshorn et al.’s chapter on behavior modeling in AI systems solutions. This learning function could be integrated in conjunction with upgrading the Graphical User Interface (GUI). The researcher would need to update and re-engineer the GUI to make it more interactive and easily configurable, perhaps using a different software application or programming language, such as MATLAB. This re-engineered GUI would allow for a much more dynamic operator interface to customize behaviors, update behaviors, and set recognition and learning conditions for the sensor processing, as well as the behavior analysis algorithms. This learning function would also need to incorporate safeguards to prevent enemy manipulation of the behavior learning processes, in an attempt to “spoof” the behavior analysis system.

A future phase in continuation of this thesis research would be implementation, testing and demonstration of further improvements to the POCS, to include incorporating automation in the updating and processing of behaviors, as well as the updating and processing of environment map data and calibrated camera data. In addition, an improved "near-real-time" video processing functionality has also been planned for implementation and testing as part of this future phase of research. It is anticipated that these improvements will serve to transition the POCS to a prototype system, as well as more readily show the scalability of the POCS to an operationally viable system.

Another very vital research topic to propel the implementation of ABA in a MDA, as well as other, environments and mission applications, would be developing a mobile platform that hosts the ABA capability. Much work is currently being done in the field of automated robotics in the Network-Centric System Engineering (NCSE) Lab within the NCSE Track at NPS. This research could be expanded to include hosting a small processing capability on one or more robotic agents that would then require the capability to autonomously orient and update their surroundings in real time to understand behavior patterns of tracks that are moving in relation to their own position in space and time. Some of this work has already begun in the development of the Wireless Smart Sensor Network (WSSN), which self-orients and communicates its position to other agents in the network. If these evolutions were scaled and expanded to include the communication of real-time updates in calibrated sensor (camera) "map" data, the agent(s) would then have a basis from which to process and analyze track data for behavior analysis and classification. This very important step would pave the way for integrating ABA on any type of mobile platform, greatly expanding the portability and flexibility of behavior analysis to create "Smart Environments" within any Net-Centric environment.

Yet another key research field would involve improved automated cost calculation routines and methods within the BAM. Possibly a collaborative work between the disciplines of defense analysis, mathematics and computer science, but applied via SE, would be developing robust algorithms (e.g., neural networks) for learning and then applying advanced statistical methods to smart environments, which would collect and process the statistical data for various behaviors as part of the behavior pattern recognition and learning function. This learned statistical data could then be applied in automating cost calculation when determining relative "distances" from observed behaviors to stored/learned behaviors. This automated cost calculation method would greatly improve accuracy and currency of behavior analysis and relieve operators and maintainers from tedious manual updates of behavior software in a highly dynamic environment.

Lastly, to fully capitalize on the breadth of ongoing research regarding automation in MDA, NPS SE researchers should seek collaboration opportunities with current automation R&D initiatives in the MDA realm. Discussion and information sharing with the developers of such system solution projects such as Predictive Analysis for Naval Deployment Activities (PANDA), Comprehensive Maritime Awareness (CMA), and Automated Scene Understanding (ASU), could yield very promising results in launching from the successes that those initiatives have shown.   Applying the principles of DIPR to these initiatives, namely the automated feature fusion to syntactical grammar-based intelligent states (symbols) for behavior analysis, which more easily facilitates ABA integration into existing MDA networks, could offer great dividends in reducing processing and memory requirements, as well as speeding implementation of needed capabilities. This collaboration has the potential to produce very promising results in automating and enhancing many MDA functions across the spectrum of MDA environments.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

The Watchman Maritime Smart Environment Proof of Concept System Description Document may be found by going to the following: http://edocs.nps.edu/npspubs/scholarly/theses/2010/Jun/WMSE-SDDver9Davis.pdf

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Alberts, D. S., J. J. Garstka, and F. P. Stein. (2000). Network Centric Warfare: Developing and Leveraging Information Superiority, CCRP Publ., 2nd Edition (Revised). August 1999, Second Print February 2000.

Blanchard, Benjamin S. and Wolter J. Fabrycky. Systems Engineering and Analysis. 17–18, Englewood Cliffs, NJ: Prentice-Hall, 2006.

Bonasso, R. P. "What AI can do for battle management," *AI Magazine* 9, 1988, 77–83.

CJCSI 3010.02B. Joint Operations Concepts Development Process (JOpsC-DP), January 27, 2006

CMA JCTD Management Plan (Revision A). February 20, 2007, 1–1.

CMA JCTD Operational Utility Assessment Final Report, May 2009, 6.

DARPA Information Processing Techniques Office (IPTO)/Programs/Predictive Analysis for Naval Deployment Activities (PANDA), http://www.darpa.mil/ipto/programs/panda/panda_goals.asp, accessed April 8, 2010.

Department of Defense. (September 2007). "Common Operational Activity List COAL v2.0."

Department of Defense, Joint Operations Concepts, November 2003, 16.

Department of the Navy. (April 2007). "Joint and Naval Capability Terminology Lists (CMCL)."

Doggrell, Les. Operationally Responsive Space – A Vision for the Future of Military Space, *Air & Space Power Journal*, Summer 2006, available at http://www.airpower.maxwell.af.mil/airchronicles/apj/apj06/sum06/doggrell.html, accessed April 9, 2010.

Faceli, K., A. C. P. L. F. de Carvalho, and S. O. Rezende. "Combining intelligent techniques for sensor fusion," *Proceedings of the 9th International Conference on Neural Information Processing, 2002 (ICONIP '02)*, vol. 4, 1998–2002, IEEE Press, New York, Pub. No. 981-04-7524-1, 2002.

"Findings for network-centric systems engineering education." Presented at the Military Communications (MILCOM) Conference, San Diego, CA. November 2008.

"FORCEnet Status Today." Briefing to the Strike, Land Attack & Air Defense (SLAAD) Division. Space and Naval Warfare Systems Center, April 29, 2004.

"FORCEnet, A Functional Concept for the 21st Century." Admiral Vern Clark, U.S. Navy General Michael Hagee, U.S. Marine Corps, February 7, 2005.

Global Security-MH-60R. http://www.globalsecurity.org/military/systems/aircraft/sh-60r.htm, accessed April 5, 2010.

Global Security-Sonar. http://www.globalsecurity.org/military/systems/ship/systems/an-sqs-53.htm, accessed April 7, 2010.

Global Security-Maritime Domain Awareness. http://www.globalsecurity.org/intell/systems/mda.htm, accessed April 2, 2010.

Global Security-FORCENet. http://www.globalsecurity.org/military/systems/ship/systems/forcenet.htm, accessed March 25, 2010.

Goshorn, Deborah E., Goshorn, Joshua L. and Goshorn, Lawrence A. "Behavior Modeling for Detection, Identification, Prediction, and Reaction (DIPR) in AI Systems Solutions" Handbook of Ambient Intelligence and Smart Environments, Springer Handbook (http://www.springerlink.com/content/n812r0064785g764/), accessed April 18, 2010.

Goshorn, R. E. "Smart Robot Workshop" Introduction Brief, Naval Postgraduate School, April 2010.

INCOSE Systems Engineering Handbook. version 3, June 2006. Edited by: Cecilia Haskins.

J7, Joint Force Development and Integration Division (JFDID), Maritime Domain Awareness JIC, Version 0.3, January 8, 2009.

Jane's Avionics. Airborne electronic warfare (EW) systems, United States, February 3, 2010, http://www8.janes.com/JDIC/JDET, accessed April 5, 2010.

Jane's Avionics. Airborne electronic warfare (EW) systems, United States, December 8, 2008, http://www8.janes.com/JDIC/JDET, accessed April 5, 2010.

Jane's Naval Weapon Systems. September 16, 2009. http://www8.janes.com/JDIC/JDET, accessed April 1, 2010.

Jane's Radar and Electronic Warfare Systems. http://www8.janes.com/JDIC/JDET, accessed April 2, 2010.

Langford, Gary. "Systems Integration and Development." Lecture slides, Naval Postgraduate School, November 2009.

Maritime Automated Super Track Enhanced Reporting (MASTER) Joint Capability Technology Demonstration (JCTD).

Military Analysis Network. http://www.fas.org/irp/program/collect/surtass.htm, accessed April 7, 2010.

Military Analysis Network. http://www.fas.org/man/dod-101/sys/ship/weaps/an-sqr-19.htm, accessed April 7, 2010.

National Concept of Operations for Maritime Domain Awareness, December 2007.

National Oceanographic and Atmospheric Association (NOAA). VENTS Web site, http://www.pmel.noaa.gov/vents/acoustics/sosus.html, accessed April 16, 2010.

National Security Presidential Directive NSPD-41 / Homeland Security Presidential Directive HSPD-13, December 2004.

Naval Architecture Elements Reference Guide, https://stalwart.spawar.navy.mil/naerg/ accessed June 9, 2010

Naval Open Architecture Contract Guidebook, version 1.1. October 25, 2007. PEO-IWS 7.

Navy Fact File. http://usmilitary.about.com/library/milinfo/navyfacts/blsurveillanceships.htm, accessed 7 April, 2010.

Navy Tactical Task List (NTTL). NTTL 3.0 (Draft). August 1, 2004.

Net Resources International. http://www.airforce-technology.com/projects/global/, accessed April 2, 2010.

NPS Maritime Defense and Security Research. Program Newsletter, Vol. 43, February 2010.

NSSN Virginia Class Attack Submarine, USA. http://www.naval-technology.com/projects/nssn/, accessed April 7, 2010.

Office of Naval Research. Broad Agency Announcement 09-023. “Multi-Target Track and Terminate (MT3),” 2009.

Operational Demonstration 2 Operational Utility Assessment Report. April 2009, 5.

“Plan for Operationally Responsive Space.” A Report to Congressional Defense Committees, National Security Space Office (NSSO), April 17, 2007, 2, http://www.airpower.maxwell.af.mil/airchronicles/apj/apj06/sum06/doggrell.html, accessed April 8, 2010.

"Reconnaissance Satellite." The Columbia Encyclopedia, Sixth Edition. 2008. Encyclopedia.com. http://www.encyclopedia.com, accessed April 3, 2010.

Schafer, David C. A Systems Engineering Survey of Artificial Intelligence and Smart Sensor Networks in a Network-Centric Environment, Thesis, Naval Postgraduate School, September 2009.

"Sea Power 21, Projecting Decisive Joint Capabilities" Admiral Vern Clark, U.S. Navy Proceedings, October 2002, http://www.navy.mil/navydata/cno/proceedings.html, accessed April 15, 2010.

Seibert, Michael, Bradley J. Rhodes, Neil A. Bomberger, Patricia O. Beane, Jason J. Sroka, Wendy Kogel, William Kreamer, Chris Stauffer, Linda Kirschner, Edmond Chalom, Michael Bosse, and Robert Tillson. SeeCoast Port Surveillance, Proceedings of SPIE Vol. 6204: Photonics for Port and Harbor Security II Orlando, FL. April 18–19, 2006.

Skolnick, F. R., and P. G. Wilkins. (2000). "Laying the foundation for successful systems engineering." *Johns Hopkins Apl Technical Digest 21*, no. 2 (2000).

Smith, W. Thomas, Jr. "Pirates in the 21st Century," July 3, 2006. http://www.military.com/forums/0,15240,103960,00.html, accessed May 20, 2010.

SSN Seawolf Class Attack Submarine, USA. http://www.naval-technology.com/projects/nssn/, accessed April 7, 2010.

Thermal Imaging Cameras for Border Security and Coastal Surveillance. FLIR Systems, 2010. http://www.flir.com/cvs/eurasia/en/content/?id=9652, accessed April 14, 2010.

U.S. Navy Fact File, P-3C Orion. http://www.navy.mil/navydata/fact_display.asp?cid=1100&tid=1400&ct=1, accessed 5 April 5, 2010.

United States Dept. of Defense (DoD). Office of Force Transformation (OFT), *The Implementation of Network-Centric Warfare,* U.S. Government Printing Office, Washington, D.C., 2005.

Winstanley, G. "Artificial intelligence support for systems engineering," *IEE Colloquium on IT Support for Systems Engineers,* p. 6/1–6/5, IEEE Press, New York, Pub. No. 3844604, 1990.

Womersley, Mark. Protecting Coastal Communities through Civil Maritime Surveillance. http://www.gisdevelopment.net/application/nrm/coastal/mnm/art_001pf.htm, accessed April 7, 2010.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. CAPT Jeffrey Kline (Ret.)
   Naval Postgraduate School
   Monterey, CA

4. RADM Terry J. Benedict
   PEO Integrated Warfare Systems
   San Diego, CA

5. RDML Jerry Burroughs
   PEO Command, Control, Communications, Computers and Intelligence (C4I)
   San Diego, CA

6. RDML William E. Shannon
   PEO Unmanned Aviation and Strike Weapons
   Patuxent River, MD

7. CAPT Robert Parker
   Program Manager, PMW 120, PEO C4I
   San Diego, CA

8. Mr. Andy Farrar
   Special Assistant for Maritime Domain Awareness, PEO C4I
   San Diego, CA